Manual

Web-IO Digital Logger
6x Input, 6x Output

Release 3.10, August 2009
Type 57650

Firmware 3.10 and higher

W&T

© 12/2003 by Wiesemann und Theis GmbH
Microsoft, MS-DOS, Windows, Winsock and Visual Basic
are registered trademarks of Microsoft Corporation

Subject to error and alteration:

Since it is possible that we make mistakes, you mustn't use any of our statements without verification. Please, inform us of any error or misunderstanding you come about, so we can identify and eliminate it as soon as possible.

Carry out your work on or with W&T products only to the extent that they are described here and after you have completely read and understood the manual or guide. We are not liable for unauthorized repairs or tampering. When in doubt, check first with us or with your dealer.

Content

1. Initial overview of the functions

The W&T Web-IO Digital product family makes it possible to monitor and set digital inputs and outputs over an Ethernet-TCP/IP network. The Web-IO Digital Logger also stores up to 1 million events resp. changes to inputs, outputs, counters as well as discrete alarms.

1.1 The communication access

The Web-IO Digital Logger provides the user with an RJ45 Ethernet port for data exchange.

The RJ45 socket allws the Web-IO Digital Logger to be connected directly to the local network using a patch cable. The Ethernet connection is implemented using 10/100Mbit autosensing.

The TCP/IP-Ethernet port makes it possible to fully configure the Web-IO Digital Logger both at initial startup as well as at a later point.

Configuration can be done manually using Web-Based Management in the browser window and loaded via XML file. SNMP-supported configuration is also possible.

The complete functional scope of the Web-IO Digital Logger is also usable via the Ethernet port.

1.2 Possible operating modes - An overview

After successfully configuring the TCP/IP parameters, the W&T Web-IO Digital Logger provides a wealth of different operating modes, functions and services.

The functions can be divided into groups:

- Basic Operating Modes
- Alarm Service
- Input/Output Service
- System Service

1.2.1 Basic operating modes

Five different operating modes and communication paths can be used for setting and monitoring the inputs and outputs.

You must specify for each individual output in which mode you want it to operate. To prevent access conflicts, the output is only available for this mode and cannot be accessed in any other way.

The inputs and counters on the other hand can be read equally in all operating modes.

HTTP - browser access
This mode can be used immediately without additional configuration using the factory defaults. Browser access to the outputs is however only possible with Administrator or Operator rights. Java applets can be incorporated into your own Web pages for a current display of the input states.

TCP socket strings (ASCII)
In this mode the inputs and outputs can be controlled by sending command strings via TCP. TCP access can also be used without additional configuration using the factory defaults.

UDP socket strings (ASCII)
Just as with TCP socket string mode, the inputs and output can be controlled by sending command strings. Access via UDP must first be enabled by the administrator.

SNMP
The inputs and outputs can be controlled using a private MIB. The configuration of the Web-IO Digital Logger can also be affected. Access to the outputs must first be activated for this mode.

TCP/UDP-Binary I and Binary II
These modes allow fast, flexible socket access, with communication handled by exchanging binary structures.

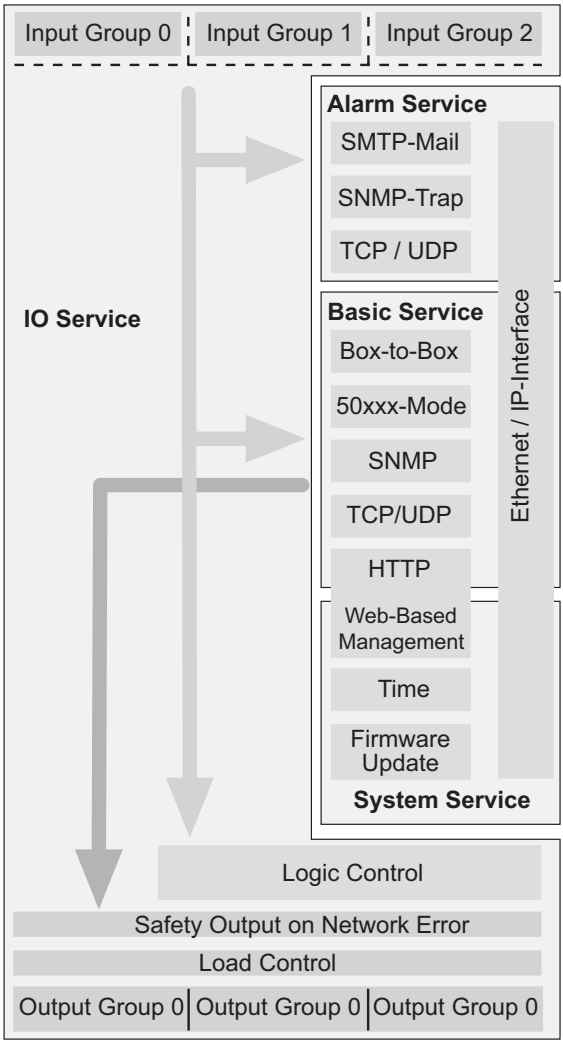
For this the Web-IO Digital Logger provides two independent sockets (Binary I and Binary II), which can be operated as TCP server, TCP client or UDP peer. It is possible to incorporate password protection when opening a connection to protect the application from undesired access by third parties.

In addition to standard client and server applications, the Web-IO Digital Logger offers three other application modes in binary mode:

- **Box-to-Box**
  In this mode the inputs on Web-IO Digital Logger correspond to the outputs of a second and vice-versa.

- **50xxx-compatibility mode**
  To operate the Web-IO Digital Logger with applications that have been programmed for W&T Digital I/O Com-Servers (predecessors to the Web-IO 1xDigital), a compatibility mode is available. Access in this mode is possible without password protection.

- **OPC**
  In OPC mode the Web-IO Digital Logger allows access from process and visualization systems through the Wiesemann & Theis OPC server, which can be downloaded free of charge from our Web site.



The block diagram above shows an overview of the various services and modes.

1.2.2 Alarm service

Along with the basic operating modes, 12 various alarms can be defined in the Web-IO Digital Logger The following alarm conditions can be configured for each individual alarm:

- Certain patterns of input and output states
- Counter states
- Overload errors
- Timer interval (for cyclical alarming)

The following communication paths are available for the alarm output:

- SMTP (E-mail)
- SNMP trap
- SYSLOG
- FTP data logging
- UDP
- TCP

1.2.3 Input /Output service

In addition to the physical properties of the inputs and outputs (See section *Connections, operating elements and LEDs*) the Web-IO Digital Logger has logic and monitoring functions.

Output groups
The outputs can be divided into groups of two or four to increase the maximum switching current of 500mA. *(not applicable to Web-IO Relays)*

Outputs that work in a group always switch at the same time, even if only one output was triggered! This makes it absolutely essential to use the outputs in a group using the same mode.

Logic functions
The individual outputs can be controlled as a function of configurable input patterns.

One example: Output 11 can be configured so that it is set to ON when Input 0 = ON and Input 1 = Off. If this condition is not met, Output 0 = OFF.


Outputs that were configured for such a logic function can no longer be controlled by the basic operating modes.

Safety Output
A safety pattern can be defined for the outputs. If the Web-IO does not detect any network activity during a configurable timeout, the outputs are set to this safety pattern. As son as triggering from the network side resumes, the outputs are set to their original state and can be controlled normally.

Load Control
The outputs have thermal protection and are monitored by the Web-IO Digital Logger for overload or the absence of a load. In case of error the outputs are turned off. *(not applicable to Web-IO Relays)*

1.2.4 Data Logger

The Web-IO Digital Logger has 8 MB of internal memory. Here changes to inputs, counters and outputs as well as discrete alarms are stored with a time stamp.

To be able to detect all changes, a time interval of at least 100ms must be kept between two events.

If for example changes occur on the inputs at a faster rate, an image of the counter states is stored after 100ms. This means you can at least detrmine how often the signals have changed on the individual outputs.

The memory is sufficient for approx. a million events and operates using the ring memory principle. Once the entire memory is full, the oldest events are overwritten.

There are three ways to read out the logger or discrete ranges:

- Download as Excel-compatible CVS file
- Page-by-page display in the browser
- Reading the data records from individually programmed applications using TCP/IP sockets.

1.2.5 System Service

In addition to the Web-IO Digital Logger functions that directly access inputs and outputs, there are a few additional important system functions. These include:

- Web-Based Management
- System clock incl. time server synchronization
- Capability for firmware update

All the functions briefly described here are covered in greater detail in the following sections.

2 Before starting up

We recommend to read the complete Manual for the Digital Web-IO Digital Logger before startup.

- Connecting the supply voltage
- Connecting to the network
- Assigning the IP address

2.1 Supply voltage and network connection

Only a few steps are necessary to incorporate the Web-IO Data Logger into a network an place it in service.

2.1.1 Supply voltage

In the following the two methods of supplying the Web-IO Digital Logger with power are derscribed.

The types of supply voltage described here only power the device itself. The in- and outputs require their own power supply.

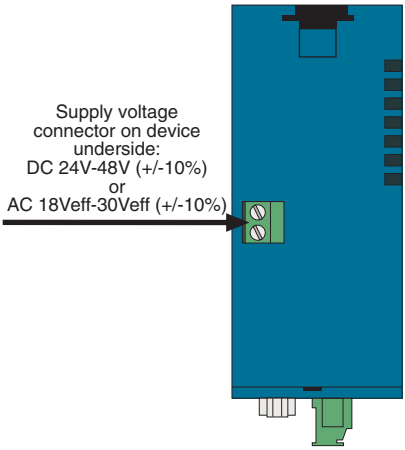
If the device is powered with the necessary supply voltage via PoE, connecting or disconnecting an additional external power supply while running may result in a restart of the Web-Alarm. The device does resume its service automatically after a few seconds, but IO events which may occur in the meantime will not be stored in the internal logger.

2.1.2 External supply voltage

Connect 24V...48V DC (+/-10%) or 18Veff...30Veff AC (+/-10%) to the terminal on the underside of the unit. You may use W&T power supplies or any power supply which meets the technical requirements.

The external power supply for the unit is always required in networks without PoE support, but may also be used in PoE environments.

When powering with DC voltage, note the correct polarity. The screw terminals on the green supply voltage connector are correspondingly labeled.



Underside with terminal for external power supply

It is also possible to power the unit with 12V DC. Here however you should take into account the very poor efficiency of the power supply and the associated increased current dreaw.

2.1.3 PoE

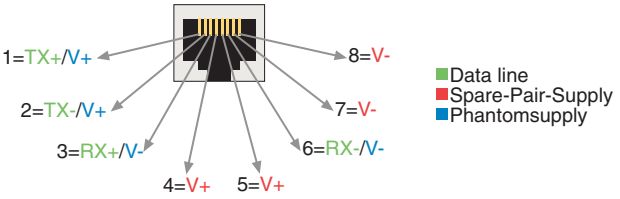
TheWeb-IO Digital Logger is equipped for use in Power-over-Ethernet environments per IEEE802.3. Here the supply voltage is brought in through the network infrastructure using the RJ45 terminal. The device supports both phantom power using data pairs 1/2 and 3/6 or Spair-Pair power on the unused wire pairs 4/5 and 7/8.

To provide power management for the supplying components, the Web-IO Digital Logger identifies itself as a Power Class 1 device with a power consumption of from 0.44W to 3.84W.

The Web-IO Digital Logger can be used with aneexternal power supply in networks not having PoE support.

2.1.2 Connecting to the network

The Web-IO Digital Logger has an IEEE 802.3 compatible network terminal on a shielded RJ45 connector. The wiring conforms to an MDI interface (see illustration), so that the connection to a hub or switch is made using a 1:1 wired and shielded patch cable.



Wiring the RJ45-POE network terminal

Thanks to the integrated Power-over-Ethernet technology the device can be powered with the required operating voltage through the network terminal.

2.2 Assigning the IP address

The Web-IO Digital Logger is factory set to IP address 0.0.0.0. Before assigning a new one, you must obtain an appropriate IP address from your system administrator. In smaller, unrouted networks use the IP address of your PC and simply change the last digit.

⚠ Always bear in mind that IP addresses must be unique within a network.

- Assigning using the WuTility tool
- Assigning using the ARP command
- Assigning through the serial port
- Assigning via RARP protocol
- Assigning via DHCP/BOOTP protocol

2.2.1 Assigning the IP address using the Wutility-Tool

If no PC with a Microsoft Windows operating system and installed TCP/IP stack is available, you cannot use this method. In this case continue starting with section 2.2.2.

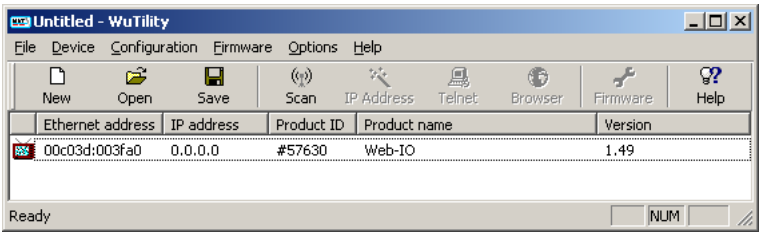
Insert the W&T Product CD included with the Web-IO Digital Logger. If the CD does not automatically start, run *cdstart.exe in the master directory by double-clicking on it.*

Select *Wutility / Install.*

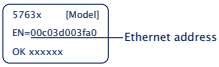
Start WuTility by double-clicking oin the *Easy Start* icon on the desktop.

Wutility now automatically scans your network for available W&T devices.

The list of found W&T network components should now contain an entry for the Web-IO Digital Logger. The IP address should be indicated as 0.0.0.0 if you are starting up for the first time. Use the displayed Ethernet address to ensure that the entry corresponds to your Web-IO Digital Logger.

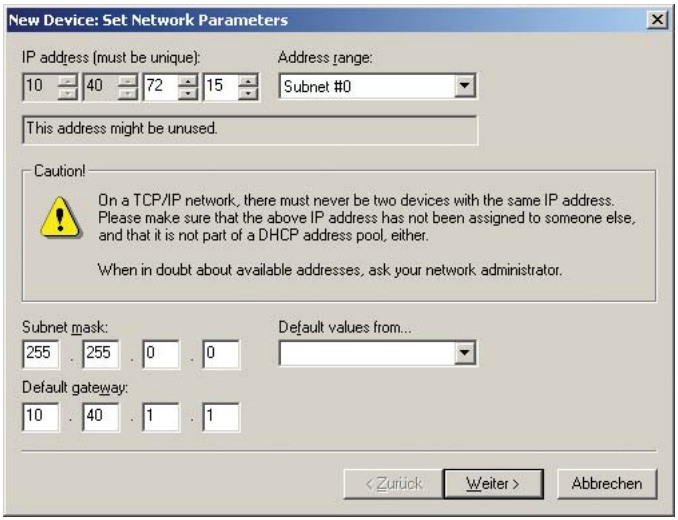


Read the Ethernet address of the Web-IO Digital Logger from the label on the side of the housing:



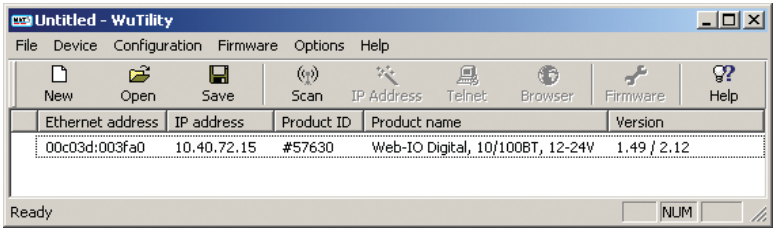
Highlight the entry for your Web-IO Digital Logger.

Click on the IP address icon and enter the IP address you want to assign to your Web-IO Digital Logger.



Check whether the entries for subnet mask and gateway are correct and confirm with *Continue*. Once all the settings have been correctly added, the Web-IO Digital Logger appears with the desired IP address in the list.

After scanning the network again Wutility should update the display in full.



All other settings can now be made conveniently from the browser using Web-Based Management.

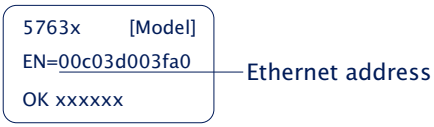
After loggin in as Administrator, use the mode *Profiles* to configure standard application cases. Detailed descriptions for configuring the various modes can be found in the following sections.

On our Web site www.WuT.de you will see at left „Search by article no.“. Enter here the Article No. of your Web-IO Digital Logger, select „Tools“ in the field below and click on „Go“. On the page which then opens select the link „WuTility“.

2.2.2 Assigning the IP address using the ARP command

This method can only be used if the Web-IO Digital Logger does not yet have an IP address, i.e., if the entry is 0.0.0.0. To change an IP address, use one of the other methods described in this section or use the menu item Config/ Device/Network in the Web-Based Management of the Web-IO Digital Logger.

An additional prerequisite is a computer which is located in the same network segment as the Web-IO Digital Logger and on which TCP/IP protocol is installed. Read off the Ethernet address of the Web-IO Digital Logger from the sticker on the side of the housing:



Now use the following command to add a static entry to the ARP table of your PC :

```
arp-s[IPaddress] [MACaddress]
```

Example under Windows:

```
arp -s 10.40.72.15 00-C0-3D-00-3F-A0
```

Example under SCO UNIX:

```
arp -s 10.40.72.15 00:C0:3D:00:3F:A0
```

Older Windows systems accept a static entry only if a dynamic entry already exists. Here you should first ping another network station.

Finally start the Web browser and enter the following URL:

http://[IP address] [*Return*]

Example: 

In Windows environments IP addresses must be entered without leading zeroes. Otherwise the system interprets the entry incorrectly and an incorrect IP address is assigned to the Web-IO Digital Logger.

The Web-IO Digital Logger takes the IP address of the first network packet sent to its Ethernet address as its own and saves it in non-volatile memory. The HTTP connection is then opened and the Homepage of the Web-IO Digital Logger appears in the browser. All other settings can now be easily made via Web-Based Management (see sec*tion Basic setting of the network parameters*).

2.2.3 Assigning the IP address through the serial port

After a reset of the Web-IO Digital Logger a time window of approx. 1-2 seconds is made available, during which entering at least 3 „x“ makes it possible to assign a new IP address.

In contrast to the previously described method via ARP, the serial way works regardless of whether the Web-IO Digital Logger already has an IP address or not. The procedure may be repeated as often as desired. Therefore you should use this method if you don‘t know the IP address or have forgotten it. The appendix to this manual contains a detailed description for using Windows with Hyperterminal.

First connect the Web-IO Digital Logger to the serial port of a computer. For a standard PC a *crossed* RS232 cable (= null modem cable) is required (see section *Serial connection*).

The serial transmission parameters of the terminal program used are configured for *9600 baud*, *no parity*, *8 bits*, *1 stop bit*, *no handshake*. Press the reset button on the Web-IO Digital Logger briefly and enter the letter „x“ at least three times on the terminal until the prompt *IPno.+<Enter>:* appears in the terminal program.

Use the conventional format (xxx.xxx.xxx.xxx) to enter the IP address, and finish your entry with *<Enter>*. If the entry was accepted, the system acknowledges with the assigned IP address. Otherwise a *FAIL* message is issued followed by the last current IP address.

All additional settings can now be made easily via Web-Based Management in the browser (see section *Basic setting of the network parameters*).

Serial assignment of subnet mask and gateway

Together with the IP address, the subnet mask and gateway address can also be assigned serially. The entry is comma delineated following the IP address. The following example assigns IP address 10.40.72.15, subnet mask 255.255.0.0 and gateway 10.40.1.1 to the Web-IO Digital Logger.

```
IP no.+<ENTER>:                          <- Web-IO Digital Logger
10.40.72.15,255.255.0.0,10.40.1.1        -> Web-IO Digital Logger
10.40.72.15 ,255.255.0.0,10.40.1.1-1     <-Web-IODigitalLogger
```

Serial deactivation of the DHCP-/BOOTP client

The DHCP-/BOOTP function of the Web-IO Digital Logger can be turned off while serially assigning the IP address. We recommend doing this in any case where the IP assignment will not be expressly performed using DHCP/BOOTP.

To deactivate the DHCP/BOOTP client, enter the option „-0“ (zero) immediately following the IP address (no spaces!) and finish the entry with *<Enter>*.

Example:

```
xxx                                    -> Web-IO Digital Logger
IP no.+<ENTER>:                        <- Web-IO Digital Logger
10.40.72.15,255.255.0.0,10.40.1.1-0    -> Web-IO Digital Logger
10.40.72.15,255.255.0.0,10.40.1.1-0    <-Web-IODigitalLogger
```

This of course functions the same way when entering the subnet mask and gateway..

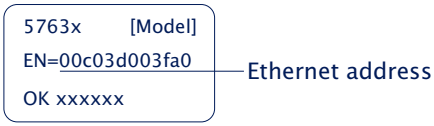
You can reactivate the function at any time later by using Web-Based Management im the browser.

Menu item: *Config >> Device >> Network*

2.2.4 IP assignment via RARP server

UNIX environments especially often use RARP protocol for centrally assigning IP addresses. Here TCP/IP devices that want an IP address send RARP requests with your Ethernet address as a network broadcast.

Activate the RARP server on the UNIX system and enter the Ethernet address of the Web-IO Digital Logger in the file */etc/ethers*. You must also enter the IP address of the Web-IO Digital Logger in the file */etc/hosts*.



The Web-IO Digital Logger must be located in the same subnet as the RARP server.

Example
Your Web-IO Digital Logger has the MAC address EN= 00C03D003FA0 (device sticker). You want to give it the IP address 172.16.232.17 and the alias name WEBIO_1:

- Entry in the file */etc/hosts*:
  `10.40.72.15 WEBIO_1`
- Entry in the file */etc/ethers*:
  `00:C0:3D:00:3F:A0 WEBIO_1`

If the Web-IO Digital Logger's IP address is 0.0.0.0 (=Factory Default), RARP broadcasts are cyclically generated in order to obtain a valid address from any existing RARP daemon.

If the Web-IO Digital Logger already has a valid IP address, an RARP broadcast is generated following each reset. If a reply follows within 500ms, the IP address contained therein is accepted. As with assigning via the serial port, this method also makes it possible to overwrite a current IP address.

2.2.5 Assigning the IP using DHCP protocol

Many networks use DHCP (Dynamic Host Configuration Protocol) or its predecessor described in the following section for centralized and dynamic assignment of the network parameters. DHCP protocol is activated by the factory default settings, so that in network environments dynamic IP assignment is sufficient for connecting the Web-IO Digital Logger to the network. The following parameters can be assigned using DHCP:

· IP address
· Subnet mask
· Gateway address
· DNS server
· Lease time

To prevent unintended address assignments or changes, we recommend deactivating the DHCP, BOOTP and RARP protocols if they are not expressly used in the respective network environment. Web-IO Digital Loggerss with incorrectly assigned IP addresses can be easily found after the fact using the scan function of the WuTility management tool and reconfigured.

Activating/Deactivating DHCP
The factory default setting is for DHCP protocol active. To deactivate it or to enable it again later, use one of the following options.

- **Management-Tool WuTility**

Select the desired Web-IO Digital Logger in the device list and click on the IP Address button. In the first dialog box enter the newly assigned network parameters and then click on Continue. In the following dialog BOX deactivate the options BOOTP and DHCP. Clicking on Continue then sends the new configuration data to the Web-IO Digital Logger.

- **Serial port**

As part of serial IP assignment, the following options for deactivating/activating DHCP and BOOTP can be selected directly following the address string:

-0 → Deactivates DHCP and BOOTP
-1 → Activates BOOTP/RARP
-2 → Activates DHCP

A detailed description of the procedure can be found in the section on *Serial assignment of IP, subnet mask and gateway*.

- **Web Based Management**

In the menu branch *Config >> Device >> Basic Settings >> Network -> DHCP/BOOTP Enable* you can alternatingly enable or disable both protocols. For detailed information refer to the section *Configuring the network parameters.*

System Name
To support any automatic updating of the DNS system by the DHCP server, the Web-IO Digital Logger identifies itself within the DHCP protocol with its system name. The factory default setting for this is *WEBIO_* followed by the last three places of the Ethernet address. For example the factory set system name of a Web-IO Digital Logger with the Ethernet address 00:c0:3d:01:02:03 is *WEBIO_010203*. The system name of the Web-IO Digital Logger can be changed in the configuration. For additional information refer to the section *Menu: Config >> Device >> Text*

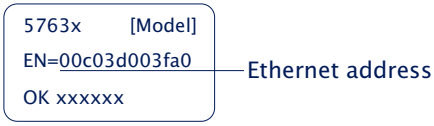
Lease-Time
The lease time determined and transmitted by the DHCP server specifies the Time-To-Live of the assigned IP address. After half the lease time has expired, the Web-IO Digital Logger attempts to extend the time for the assigned DHCP server and up update the address. If this is not possible by the time the lease time expires, for example because the DHCP server can no longer be reached, the Web-IO Digital Logger deletes the IP address and starts a new cyclical search for alternate DHCP servers for the purpose of assigning a new IP address.

Because of the absent clock, the lease time associated with the current IP address is no longer available after a reset. After the restart therefore a corresponding update request is issued with the original DHCP server. If the latter is not resolvable at this point in time, the Com-Server deletes the IP address and starts a new cyclical search for alternate DHCP servers.

If DHCP is activated, the remaining lease time together with the current IP address is displayed in the menu item *Config >> Device >> Basic Settings >> Network* using the format hh:mm:ss.

If after the assigned lease time has expired the DHCP server is not reachable, the Web-IO Digital Logger deletes its IP address. All existing TCP/UDP connections between the Web-IO Digital Logger and other network clients are thereby closed. To prevent such events, we recommend configuring the assigned lease time in the DHCP server to infinite if possible.

Reserved IP addresses
If the Web-IO Digital Logger is used as a TCP server or UDP peer, it provides services which other clients in the network can also make use of as needed. To open a connection, they of course need the current IP address for the Web-IO Digital Logger, so that in such situations it makes sense to reserve a particular IP address for the Web-IO Digital Logger on the DHCP server. This is generally done by linking the IP address to the unique Ethernet address of the Web-IO Digital Logger, which can be found on the sticker attached to the housing.



Dynamic IP addresses
Fully dynamic address assignment, whereby the Web-IO Digital Logger gets a different IP address every time it is restarted or after the lease time has expired, only makes sense in network environments with automatic cross-connection between the DHCP and DNS services. This means when a new IP address is assigned to the Com-Server, the DHCP server then automatically updates the DNS system as well. The new address is associated with the respective domain name. If in doubt, consult your system administrator for detailed information about your network environment.

If the Web-IO Digital Logger is configured as a TCP or UDP client and itself actively searches for a connection to server services in your network, dynamic changing IP addresses may be used.

2.2.6 Assigning the IP using BOOTP protocol

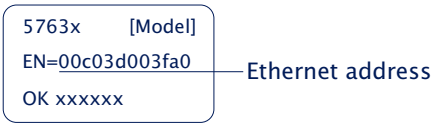
Some networks use BOOTP as predecessor of DHCP protocol for centralized and dynamic assignment of IP addresses. The factory default setting is for BOOTP turned off. You can activate it from *Config >> Device >> Basic Settings >> Network*. The following parameters can be assigned:

- IP address
- Subnet mask
- Gateway address
- DNS server

To prevent unintended address assignments or changes, we recommend deactivating the DHCP, BOOTP and RARP protocols if they are not expressly used in the respective network environment. Web-IO Digital Loggers with incorrectly assigned IP addresses can be easily found after the fact using the scan function of the WuTility management tool and reconfigured.

Address reservation

BOOTP protocol is based on fixed reservations of fixed IP addresses for particular Ethernet addresses. This means a Web-IO Digital Logger connected to the network only gets an IP address if the latter was previously stored in the BOOTP server. Check with your system administrator for creating this reservation. The Ethernet address of the Web-IO Digital Logger can be found on the housing sticker.

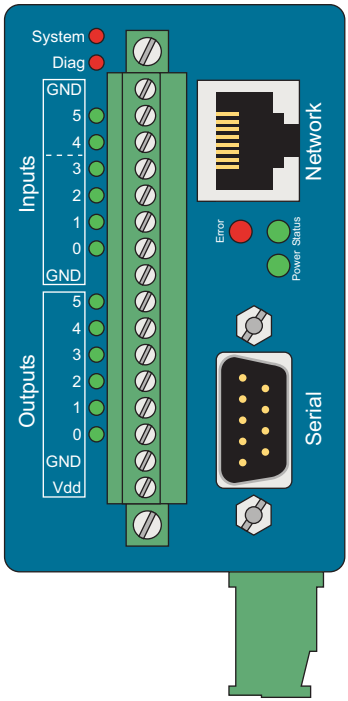


Once the administrator has made the necessary entries, the Web-IO Digital Logger obtains the desired IP address automatically after each reset. To ensure accessibility of the Web-IO Digital Logger even should the BOOTP server go down, the previous IP address is retained should there be no reply.

3 Connections, operating elements and LEDs

- Supply voltage connection
- Input and Output connections
- Ethernet connection
- RS232
- Device status LEDs

3.1 Connecting Web-IO Digital Logger



3.1.1 Supply voltage input (Vcc)

If you are not using PoE to power the device, an external power supply can be connected to the screw terminal on the underside of the device.

The power supplies available from W&T are equipped with a special connector for this on the low-voltage side.

Alternately an existing DC voltage between 24V and 48V (+/-10%) or an AC voltage between 18Veff and 30Veff (+/-10%) can be connected. Note the correct polarity when using DC voltage.

To prevent cable break, we recommend fitting the wire ends with end ferrules.

The Vdd terminals on the front side power only the switching outputs and require a DC voltage. Please read the section on Output Wiring

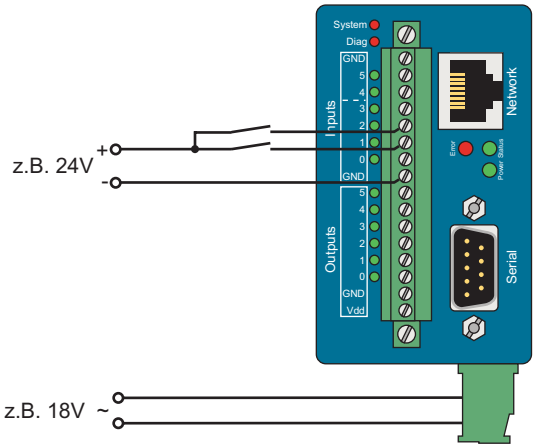
3.1.2 Input wiring

The permissible input voltage range is +/- 30V referenced to ground.

The switching threshold of the inputs is 8V +/-1V. Lower voltages are interpreted as an OFF or 0 signal. Voltages over 8V are seen by the Web-Alarm as an ON or 1 signal. Input voltages between 7V and 9V should be avoided, since they cannot be unambiguously interpreted.

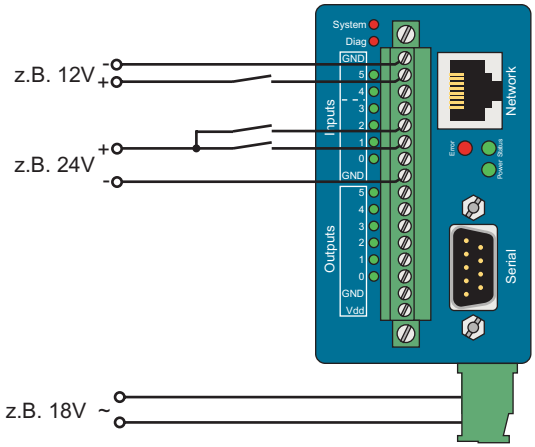
The Web-IO Digital Logger inputs are divided into two groups: Inputs 0 - 3 and Inputs 4 + 5. These groups are galvanically isolated from each other. Each input group has a separate outgoing reference ground.

The following wiring example shows how two inputs are driven. It is important that the signals for an input group have the same ground reference.



Driving two inputs in a group

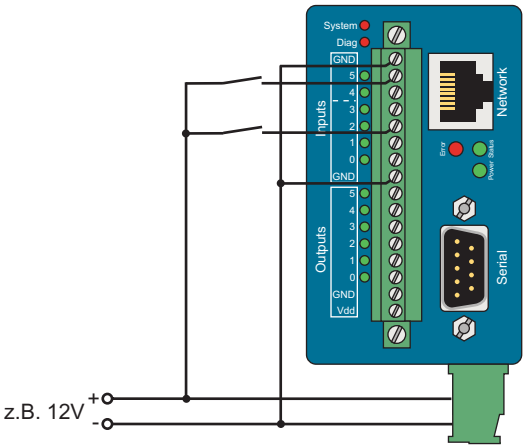
If there is a different reference ground for the input signals, these should be distributed between the two input groups.



Signals with different reference ground

If the inputs are used to monitor the states of potential-free contacts, the supply voltage for the Logger can be used as the signal voltage. In this case you must operate the Web-IO Digital

Logger with a DC voltage of 12-30V. The following illustration shows a corresponding wiring example.



Supply voltage as signal voltage

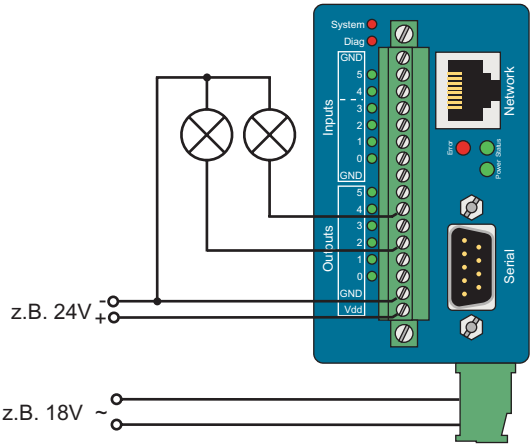
In addition to detecting the input status ON and OFF, each input also has a counter. The factory default is for counting pulses (positive edges). The counters can alternately be reconfigured from pulse to edge counting. In addition, two inputs can be used together for an incremental, direction-dependent count. Here the counter which was first recognized at the change is incremented.

3.1.3 Output wiring

The six outputs of the Web-IO Digital Logger are current sourcing. The supply voltage for the outputs can lie between 6V and 30V and is brought in through the Vdd and GND terminals in the output terminal group. The maximum switching current per output is 500mA.

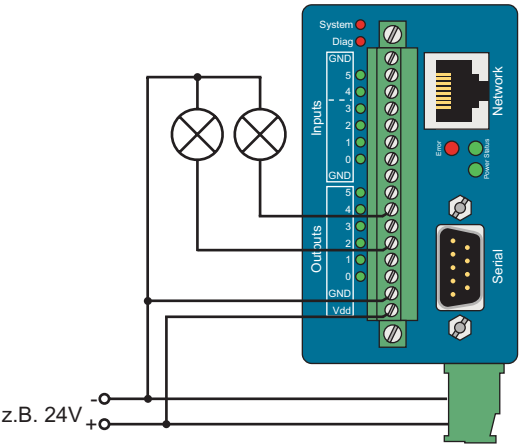
Outputs connected to a inductive load (e.g. a relay) should be protected from damage by a recovery diode.

The outputs also have thermal overload protection and are short circuit protected.



Output circuit with separate supply

When sizing the output supply voltage, you should take into account the required current. If the unit is being supplied by an external power supply with 12V-30V DC whose power is also sufficient for supplying the consumers connected to the outputs, the output supply can also be connected to the Logger supply.

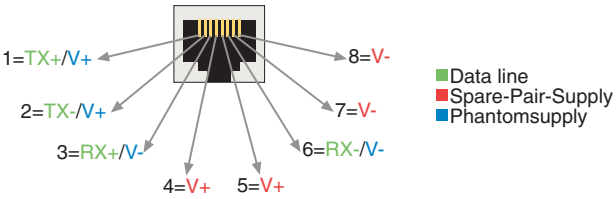


Supplying the outputs from the Logger supply

The range of the Loggger supply voltage exceeds the range of the switchable output volrtage. If you use the Logger supply for supplying the outputs as well, provide no more than 30V to the Logger.

3.2 Ethernet connection

Web-IO models feature an IEEE 802.3 compatible network interface on a shielded RJ45 connector. The pin assignments correspond to an MDI interface, so that the connection to the hub or switch is made using a 1:1 shielded patch cable.



Pin configuration for the RJ45 PoE network socket

Thanks to the integrated Power-over-Ethernet technology, the device can be powered with the required operating voltage through the network connection.

10/100BaseT, Autosensing

Both 10BaseT and 100BaseTx standard with a bit rate of 100MBit/s and the possibility of full-duplex transmission are supported. Changing between the two network speeds is automatically done by the autosensing function of the Web-IO Digital Logger depending on the possibilities of the hub or switch used. The prerequisite for operation at 100MBit/s is suitable cabling (at least conforming to Cat. 5/ISO Class D).

Both stations (Switch and Web-IO Digital Logger) should be run in auto-negotiating mode.

Galvanic isolation

The network connection is galvanically isolated with respect to the supply voltage and to the digital IOs and the serial port(s) up to at least $500V_{rms}$.

Link status

The Error LED on the front of the unit indicates the current link status: If it flashes at 1-2 second intervals, there is no connection to the hub or the connection is faulty.

Manageable switches often have special protocols (Spanning Tree Protocol, Port-Trunking, ...), as required for example for uplinks to other switches or for broadband connection of servers. These protocols are not required for connecting normal terminal devices such as the Web-IO Digital Logger, but do under some circumstances delay opening of communication considerably after a restart.

In some cases it can even happen that switches do not pass data which the Web-IO Digital Logger sends after the physical links are established (e.g. ARP requests and BootP packets)!

We recommend deactivating the unneeded protocols and functions of the switch on the port used for the Web-IO Digital Logger. Please consult the responsible network administrator.

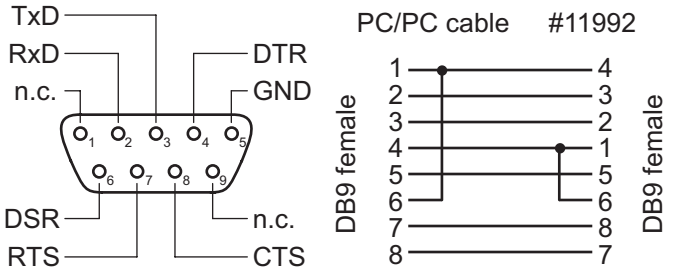
3.6 Serial connection

In the Web-IO Digital Logger 12xDigital without Com-Server functionality and in the Web-IO 24xDigital OEM the serial port is available only for configuration purposes such as IP address assignment.

The pinouts of this RS232 port are identical to those of your PC. This lets you use standard cables.

A detailed pin description as well as the W&T cables required for most common applications can be found in the illustrations below.

RS232 pin assignments and functions, DB9 plug



To serially assign the IP address, cable 1199x and 1179x are recommended.

The Web-IO 2xDigital does not have a serial port.

3.7 LEDs for device status

Power-LED:
Indicates the presence of supply voltage. If the LED does not come on, check the power supply connections.

Status-LED:
Blinks whenever there is network activity with the Web-IO Digital Logger. Periodic blinking indicates that the port has a connection to another station.

Error-LED:
The Error-LED uses various blinking codes to indicate error states on the device or network port.

1x blinking of the Error-LED
Check network connection. The Web-IO Digital Logger is not getting a link pulse from a hub/switch. Check the cable or the hub/switch port.

2x or 3x blinking of the Error-LED
Press the button on the housing to perform a reset (on the Web-IO 2xDigital by interrupting power). If this does not clear the error, reset the device to its factory defaults. Since this resets all network settings, you should write down your settings first.

Web-Based Management: *Config >> Session Control >> LogOut >> Restore Defaults* . Following a reset the device is restored to its factory defaults. Reconfigure your network settings.

More information about menus and coniguration can be found in the section Web-Based Management.

on error http://xxx.xxx.xxx.xxx/diag-LED:
Indicates internal configuration errors. For troubleshooting, go to page http://<IP address of the Web-IO Digital Logger>/diag in the unit.

For example:

```
http://172.16.232.17/diag
```

system error:
Serious hardware error. Try to restart the unit by disconnecting the power supply. If the condition persists, please return the unit to your dealer so that it can be checked by W&T.

Power-LED +Status-LED +Error-LED on = Self-test error
The self-test performed by the Web-IO Digital Logger after each start or reset could not be correctly finished, due for example to an incomplete update of the firmware. In this condition the device is no longer functional. Please return the unit to your dealer so that it can be checked out by W&T.

If the Web-IO Digital Logger has no IP address or the address is 0.0.0.0, the LEDs on error *and* system error *will remain on after a reset or new start! The LEDs will turn off only if an IP address is assigned.*

The Web-IO 2xDigital does not have the on error *and* system error *LEDs*

4 Configuring the network parameters

- Web-Based Management
- Basic setting of the network parameters

4.1 Web-Based Management

The W&T Web-IO Digital Logger Digital Logger provides the user with a complete Web-Based Management system. This means: If the Web-IO Digital Logger has an IP address, all the settings can be conveniently made over the network from the browser. Since no device-specific configuration software is needed, the user is also not tied to a particular operating system.

Regardless of whether you are using Microsoft Windows, Linux, Unix, Apple Macintosh, os2... , wherever there is a browser there is also a way to configure. And the configuration is always handled the same way regardless of the operating system.

At this point we would like to explain a few basic principles of Web-Based Management and help to understand how this works with the Web-IO Digital Logger.

4.1.1 Structure of the Web site

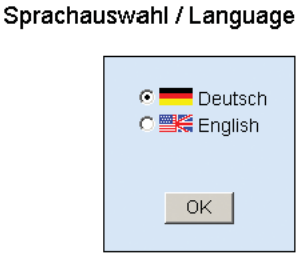
When the Web-IO Digital Logger is opened in the browser, a 2-part Web site always appears. These two parts are also known as frames.

On the left side is always the navigation frame, which one could compare with the table of contents of a book.

To the right is the display and configuration frame. When the web site is first opened, this contains the current status of the inputs and outputs of the Web-IO Digital Logger.

4.1.2 Language selection

The first time the Web site of the Web-IO Digital Logger is opened, you are prompted to select the language - German or English - you want to use for the display.



The selected language can of course be changed at any time.

It should be noted that especially in the field of computer technology many terms and concepts are commonly expressed in English. Download, firmware, update, server etc. make more sense than their German equivalents. This means that even when selecting the German version of the Web-IO Digital Logger Web site you will find a colorful mixture of German and English terms.

We hope the conversion has been done well and in a way that is fully understandable to the user.

4.1.3 Navigation

The navigation frame contains a directory tree in which all the available menu items of the Web-IO Digital Logger are listed by categories. Clicking with the mouse either displays additional menu items for a category and/or a new content is built in the configuration frame.

The icons in the menu tree have the following meaning:

Main or sub-category with branches to additional contents. Clicking on this icon expands the menu tree.

Main or sub-category with branches to additional contents. Clicking on this icon expands the menu tree and shows new contents in the configuration frame.

Indicates that additional sub-categrories or contents are present. Clicking on this icon expands the menu tree but not the contents of the configuration frame.

Clicking on this icon closes the expanded directory tree.

Designates a configuration page which is shown in the configuration frame when this icon is clicked-on.

Simply click on the corresponding country flag to switch between German and English.

Avoid using the „Forward“ and „Back“ buttons of your browser. Use only the navigation tree or buttons and links in the configuration frame to navigate. Otherwise you could lose parameters buffered in the background when you jump forward or back.

4.1.4 Easier navigation with profiles

The Web-IO 12xDigital offers a variety of operating mode and configuration possibilities. And to get even the first-time user

of the Web-IO Digital Logger off to a good start, we have created profiles for the most common applications.

After successfully loggin in as an Administrator (the next section will cover the possible access privileges), the user is given a choice between configuration using profiles and expert mode, which does not provide any configuration assistance..

Working with profiles simplifies configuration by color highlighting the configuration points which are actually needed for the corresponding mode. Both the menu entries in the navigation tree and the respective parameters on the configuration pages are highlighted.

4.1.5 Access rights

Depending on the login, the Web-IO Digital Logger distinguishes among 3 different access rights:

Default User:
Anyone opening the start page of the Web-IO Digital Logger has this status initially. The status of the inputs and outputs can be displayed but not modified. No changes to the configuration are possible.

Administrator:
Clicking on *Config* in the menu tree and entering the administrator password gives you full access to the Web-IO Digital Logger. The start page is refreshed with a significantly expanded menu tree. As shipped it is not necessary to enter a password, it is sufficient to click on the Login button.

The next sub-section describes assigning passwords in greater detail.

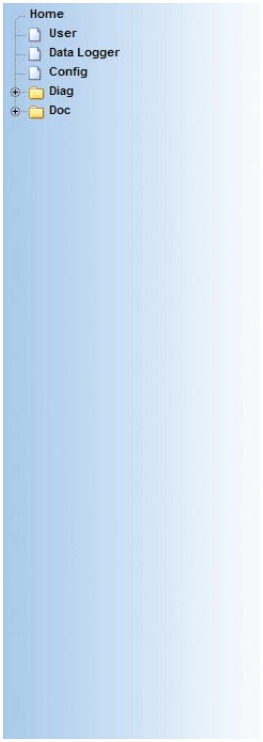
Operator:
The access rights of the operator are limited to setting the outputs, clearing the counter, changing the alarm outputs and a few diagnostic activities. The operator has no additional configuration options. The login in this mode is also

accomplished by entering the password selected for this access.

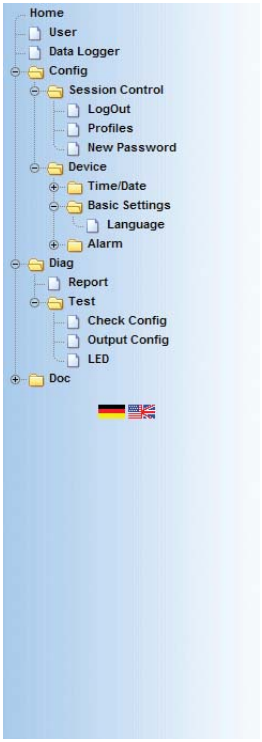
The login is done after opening the Web-IO Digital Logger using Config in the menu tree.

Here again a the menu trees for the various access rights by way of comparison:

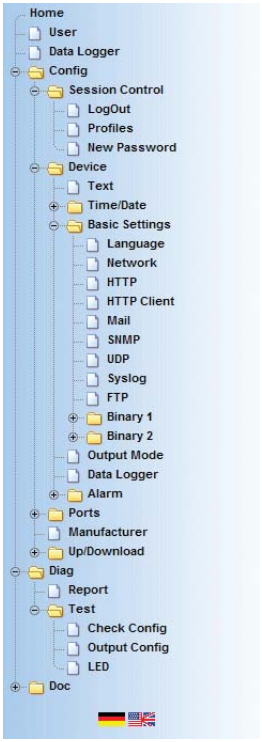
Default User



Operator



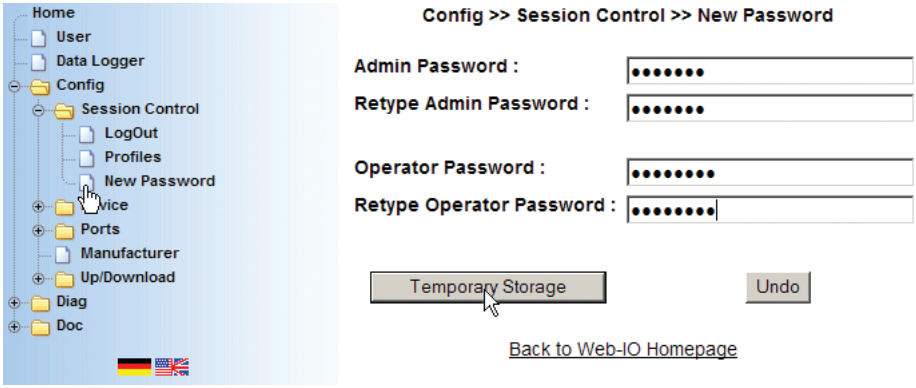
Administrator



4.1.6 Changing and creating the access passwords

The user must be logged in as an Administrator.

In the navigation tree select: *Config >> Session Control >> New Password* and enter the desired passwords in the fields. A password is limited to 31 characters.



It is not permitted to use the same password for both Admin and operator access.

Clilcking on the *Temporary Storage* button sends the new passwords to the Web-IO Digital Logger. You still have to activate the changes.

Tip: Keep a copy of the passwords in your personal documents.

If you have forgotten the passwords, see the Appendix of this manual for help.

4.1.7 Saving and activating settings

To increase you understanding of the sequences in Web-Based Management, first a few basic remarks concerning HTTP protocol:

With HTTP the browser opens a TCP connection to the HTTP server, requests Web pages and contents from the server or sends data to the server. The server then immediately closes the connection.

Entries made by the user on a Web page are first temporarily stored by the browser and only sent to the server after clicking on the *Submit* or *Temporary Storage* button through a new TCP connection.

If for example the browser is closed before the *Temporary Storage* button is clicked, the entries are irretrievably lost. This can also happen if the Web site is changed without sending the entries.

Nearly all the Web sites that can be opened from the navigation tree offer the possibility of making or changiong entries. Once you are finished with the entries on such a configuration page, you must first click on the *Temporary Storage* button and only then change to a new Web site.

Once all the entries and configuration changes have been made, the new settings must still be activated. Use the *Logout* button located on each configuration page, or use the menu path Config >> Session Control >> LogOut.

After clicking on the Save button, all the settings are updated in the Web-IO Digital Logger and the start page is refreshed in Default User mode. To make additional setting changes, you must log in again with the necessary access rights.

Login Timeout

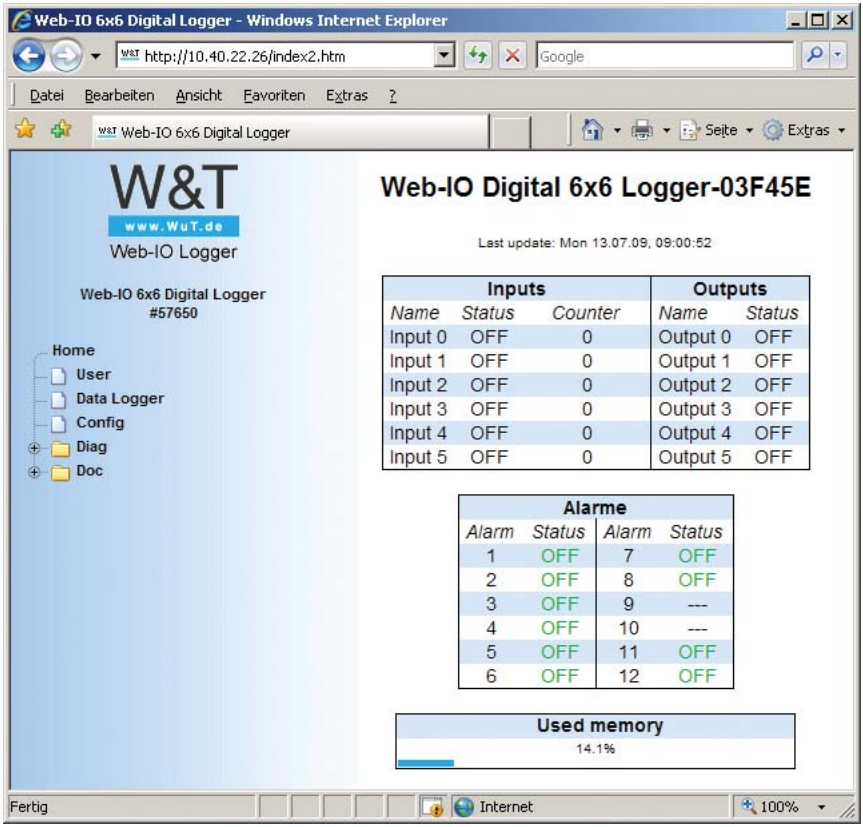
If no entry is made in Adminstrator or Operatotmode for more than 60 minutes, the Web-IO Digital Logger resets itself to Default User rights. All entries and changes made to this point, even if they were already sent to the Web-IO Digital Logger, are lost.

4.2 Basic setting of the network parameters

Whenever a Web-IO Digital Logger will be used beyond the borders of the local network, a few additional network parameters are needed in addition to the IP address.

To be able to perform the configurations described at this point, the Web-IO Digital Logger should already have an IP address for the network. If this is not the case, or if the current IP address of the Web-IO Digital Logger is unknown, first perform the steps described in section 2.

If the Web-IO Digital Logger already has an IP address, enter this in the URL field of your browser. After a few seconds the start page of the Web-IO Digital Logger will appear.



Select *Config* in the navigation menu.

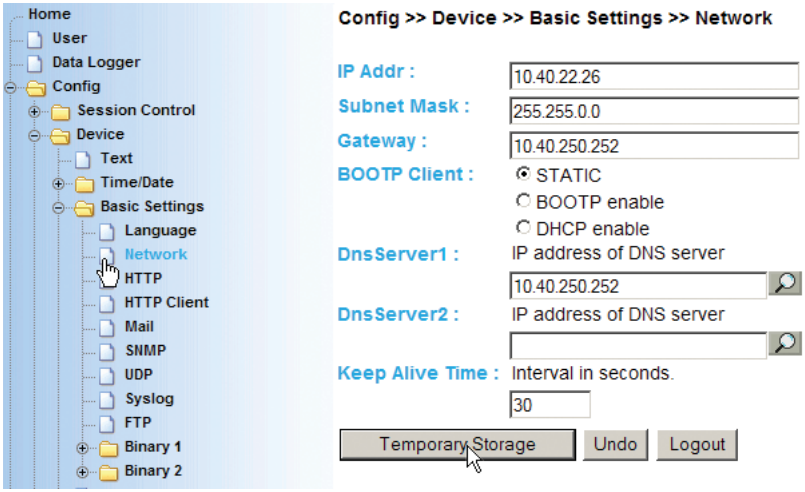
You are now prompted to enter the correct administrator password. As shipped there is no password required, and it is sufficient to click the *Login* button.

Once the correct password has been entered, the Web-IO Digital Logger confirms successful login. At this point you can choose between Expert mode and working with profiles. Use the *Profiles* button.

In the Web page which now appears select the radio button *Basic network parameters* and confirm by clicking on the *Show profiles* button.

In the navigation menu now select the color highlighted item *Config >> Device >> Basic Settings >> Network*.

Now go to the navigation menu and select *Config >> Device >> Basic Settings >> Network*. The following screen appears:



Enter here the required network parameters.

IP Addr.
The IP address may be changed at this point.

Subnet Mask / Gateway
For data exchange throughout the subnet, the appropriate subnet mask and IP address of the gateway must be entered.

DHCP Client / BOOTP Client
The factory default configuration is for the Web-IO Digital Logger to be able to obtain its IP address and other necessary network parameters from a DHCP server.

In this case the Web-IO Digital Logger receives any IP address from the address pool of the DHCP server unless a special address reservation has been made. Assured use of the Web-IO Digital Logger server services is then only possible if there is automatic synchronization in the network between the DHCP and DNS server. The Web-IO Digital Logger is then uniquely addressable using the Device Name.

Alternately, the Web-IO Digital Logger can obtain an IP address using the older BootP protocol, but this requires an address reservation on the DHCP/BootP server for trouble-free operation.

If the Web-IO Digital Logger will not be taking part in central IP address assignment via DHCP or BootP, choose static.

DNS Server
With some network partners, for example mail and time servers, it is recommended that these not be addressed using their IP address but rather using a name. A DNS server is required to resolve names into IP addresses.

If DHCP/BootP is used for assigning the network parameters, the Web-IO Digital Logger also gets the address of the DNS server automatically assigned.

The field DNS Server by DHCP is filled in by the Web-IO Digital Logger. The user can not make changes to this entry.

There are two additional fields available for manual entry of DNS servers.

To search an IP address for a name, the Web-IO Digital Logger first turns to the automatically assigned DNS server. If the latter cannot be reached or if the entry is missing, the Web-IO Digital Logger tries DNS servers 1 and 2.

Using the

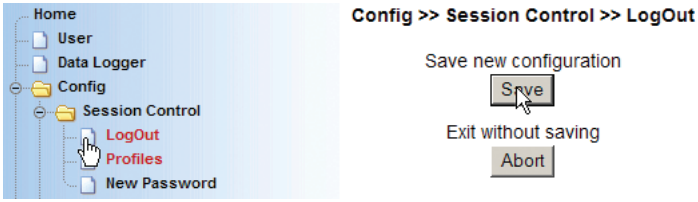
Button

you can ping the entered server. This allows you to verify whether the entered server is accessible by the Web-IO Digital Logger. If the entered server is not located in the same subnet, the condition is that subnet mask and gateway have first been stored in the Web-IO Digital Logger using Logout >> Save.

If you do not have the required information, contact your network administrator.

Keep Alive Time
During the interval (in seconds) entered here the Web-IO Digital Logger checks whether there are still connections open without actual data traffic from the client side.

When you have made all your entries, click on the *Temporary Storage* button. This sends the parameters first to the Web-IO Digital Logger. The changes do not become effective until they have been activated. For this use the *Logout* button or menu path Config >> Session Control >> LogOut.



After clicking on the *Save* button, the Web-IO Digital Logger is restarted with the current parameters. Normally the start page of the Web-IO Digital Logger is refreshed within about 10 seconds. If it does not refresh, click on the *here* link.

Note that applications which are communicating with the Web-IO Digital Logger at this point in time may experience a fault.

The Web-IO Digital Logger is now set up to the point that it can also be accessed throughout the subnet.

Changing the Device Name

By default each Web-IO Digital Logger has an individual device name which consists of the designation *WEBIO-* followed by the last 3 places of the Ethernet address.

For example, the factory set device name of a Web-IO Digital Logger having Ethernet address 00:c0:3d:01:02:03 is *WEBIO-010203*.

The device name is required especially in networks in which the Web-IO Digital Logger obtains its IP address from a DHCP server which does automatic synchronizing with the DNS system.

The Web-IO Digital Logger identifies itself within the DHCP protocol using its device name.

Using Web-Based-Management the device name can be changed at any time.

Access rights: *Administrator*

In the navigation menu select *Config >> Device >> Text.*

The following input screen appears:

Device Name

System Name with which the Web-IO Digital Logger reports to a DHCP server using DHCP protocol and which is also displayed on the homepage.

In networks with DHCP and DNS the name entered here should be unique!

Device Text

The administrator can place a text here which describes the Web-IO Digital Logger and its function. This text is displayed on the homepage of the Web-IO Digital Logger.

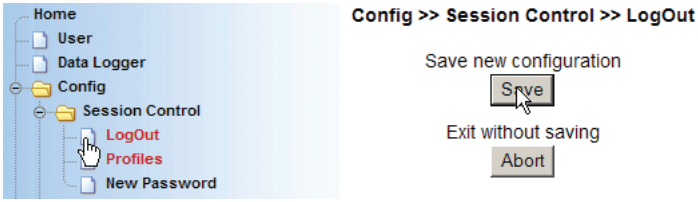
Location

Here you can enter the installed location of the Web-IO Digital Logger to make it easier to locate and maintain it in larger installations.

Contact

Here you can enter for example an e-mail address or telephone number at which the responsible technician can be reached when service is needed.

To save the texts, click on the *Apply b*utton and in the Config menu select >> Session Control >> LogOut.



After clicking on the *Save* button the Web-IO Digital Logger is restarted using the current parameters.

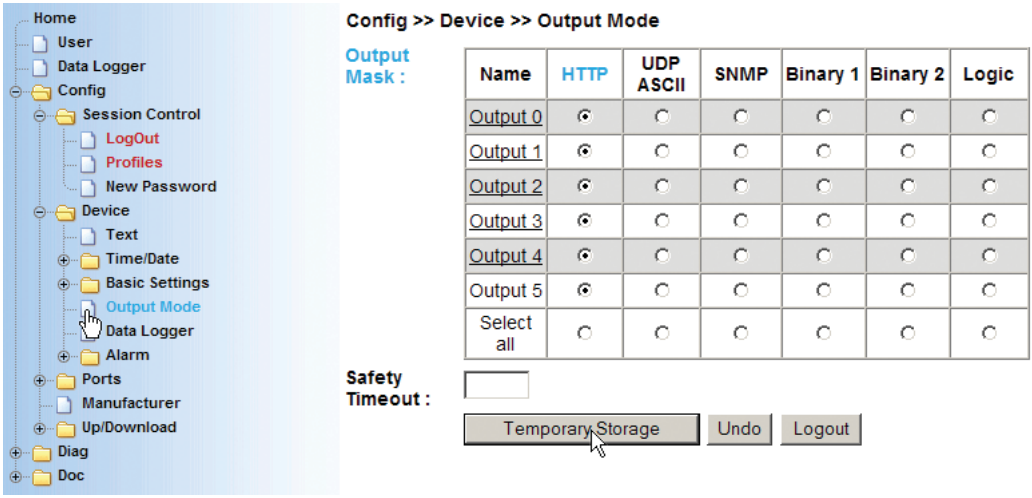
If you like to learn more about the expressions and names used, ask for a free copy of our book TCP/IP-Ethernet and Web-IO Digital Logger, where the necessary basics for starting up network components are explained. WWW.WUT.DE

5 Access types

- Overview
- HTTP - Controlling inputs and outputs in the browser
- Java Applets - Automatic refreshing in the browser
- TCP/UDP - Controlling the Web-IO Digital Logger using sockets
- Box-to-Box
- OPC
- SNMP
- Alarms

5.1 Specifying the output mode

As already described, you must specify in which mode each individual output is supposed to work. The corresponding configuration can be made under *Config >> Device >> Output Mode*.



Required access rights: *Administrator*

Output Mode

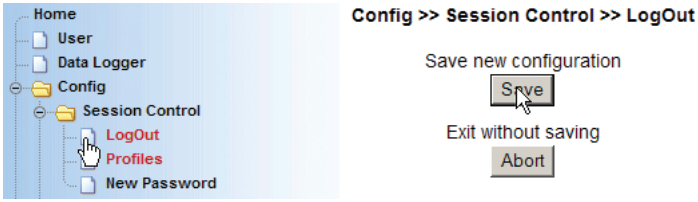
Here the mode used for each output is specified. The factory default is for all ports HTTP. Please note that in most modes a few additional settings besides output mode need to be made, such as enabling of the mode. For more detailed information, refer to the description of the respective operating mode.

Safety Timeout

If no network activity is detected for the timeout period set here, the Web-IO Digital Logger sets the outputs to a configurable output pattern. Which outputs should be involved and what their status should be is configured under *Config >> Ports >> Outputs*. A detailed description of the safety timeout and safety state can be found in the section *Safety State / Timeout*.

Load Control:
By activating this option you are enabling load monitoring. Load monitoring applies to all outputs. *(not applicable to Web-IO Digital Logger with relay outputs)*

After selecting the output modes, click on Temporary Storage to send the settings to the Web-IO Digital Logger. To activate, use the *Logout* button and click on *Save.*



For the Web-IO Digital Logger with relay outputs a pause of min. 200ms must be maintained between two switching operations on an output!

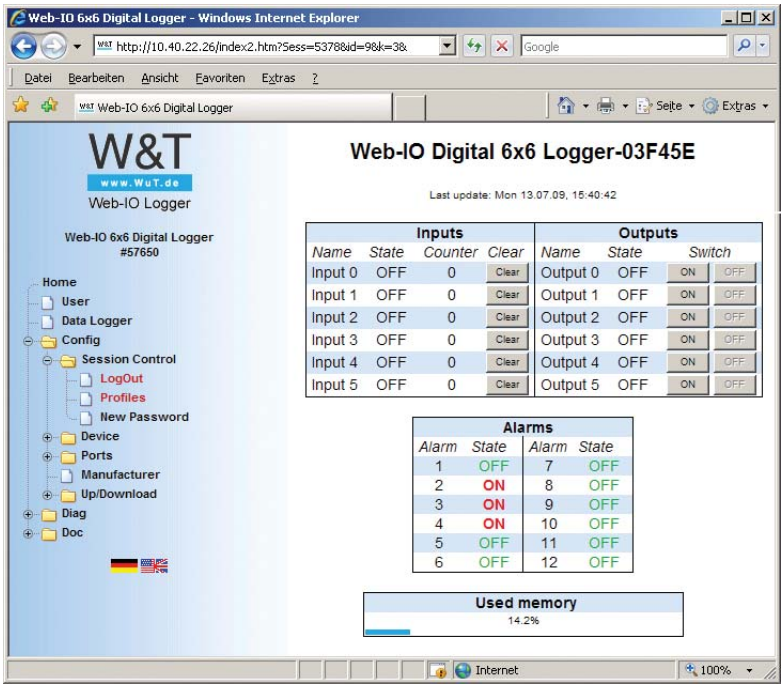
5.2 HTTP - In- and Output control from the browser

Access from the browser is the easiest way to use the Web-IO Digital Logger Digital for most users.

5.2.1 Current status of inputs, outputs and counters

As soon as the start page is opened the user is shown the status of the inputs.

- Status = ON indicates that an input voltage of >=9V was detected on the input.
- Voltages of less than 9V are indicated by status = OFF. ON status is also highlighted by a green background in the table field.
- In addition, the counter state of the individual inputs can be viewed.



To be able to control the outputs from the browser, you must log in as an administrator or with operator rights.

After a successful login, the status indications for the outputs (ON/OFF) are associated with a link. Clicking on the corresponding link changes the state on the output of the Web-IO Digital Logger and the display in the browser is updated.

The counter states of the input counters are also associated with a link; clicking on the corresponding counter sets to 0.

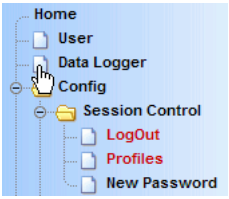
5.2.2 Logger access to stored data

All changes to inputs, outputs, counters and alarms are stored in the internal memory of the Web-IO Digital Logger. The time interval between two events must however be greater than 100ms.

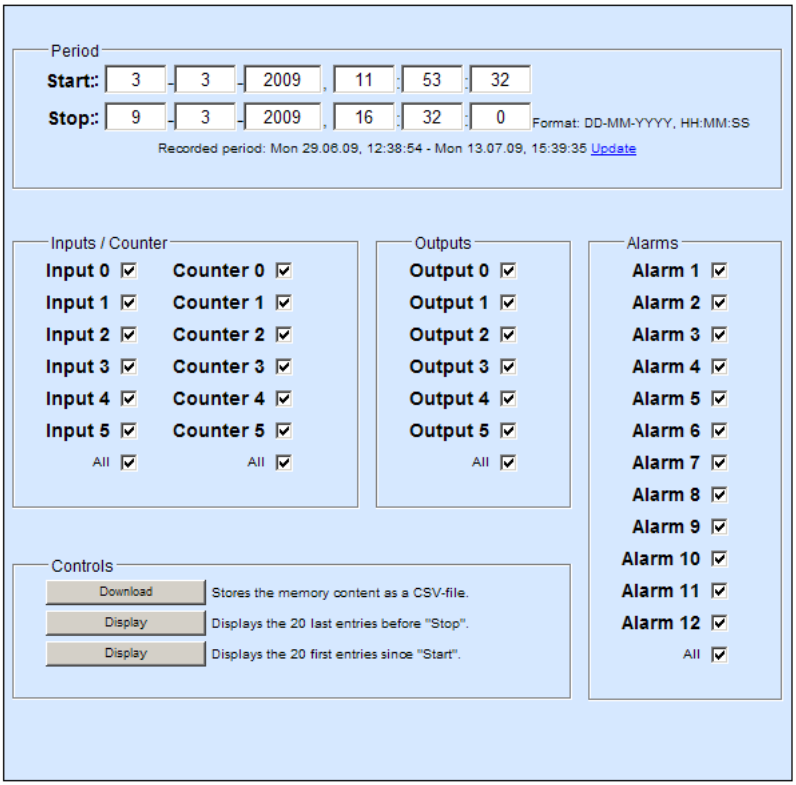
Access to the stored data may also be conveniently made from the browser.

Required access rights: *Administrator*

Select *Data Logger* from the navigation tree



Configuration page then appears for the data logger output.



First you must specify the time interval for showing and outputting the stored data.

In addition, you can also specify in advance which data you want to display. This can be specified for each individual input, counter and output as well as for the alarms.

There are three possibilities for outputting:

- The data for the selected time interval can be downloaded in entirety as a CSV file. CSV files can be opened and edited for example in Excel. (Note when selecting the time interval that Excel can show a maximum of 65535 lines.)

- The last 20 events before the specified stop time are listed in the browser in table format. If needed you can use buttons to page back.
- The first 20 events after the specified start time are listed in the browser in table format. If needed you can use buttons to page ahead.

Web-IO-03F45E
Data Logger

Datum Zeit	Inputs						Counter						Outputs						Alarme											
	I0	I1	I2	I3	I4	I5	C0	C1	C2	C3	C4	C5	O0	O1	O2	O3	O4	O5	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
Mo 09.03.09 14:54:00.100	off	ON	ON	ON	off	off	0	0	0	0	0	0	off	ON	off	ON	off	ON	off	ON	ON	ON	off	off	off	off	off	off	off	off
Mo 09.03.09 14:52:27.800	off	ON	ON	ON	off	off	0	1	1	1	0	0	off	ON	off	ON	off	ON	off	off	off	off	off	off	off	off	off	off	off	off
Mo 09.03.09 14:52:26.800	off	ON	ON	off	off	off	0	1	1	0	0	0	off	ON	off	ON	off	ON	off	off	off	off	off	off	off	off	off	off	off	off
Mo 09.03.09 14:52:26.300	off	ON	off	off	off	off	0	1	0	0	0	0	off	ON	off	ON	off	ON	off	off	off	off	off	off	off	off	off	off	off	off
Mo 09.03.09 14:47:13.400	off	off	off	off	off	off	0	0	0	0	0	0	off	ON	off	ON	off	ON	off	off	off	off	off	off	off	off	off	off	off	off
Mo 09.03.09 14:47:12.100	off	off	off	off	off	off	0	0	0	0	0	0	off	ON	off	ON	off	off	off	off	off	off	off	off	off	off	off	off	off	off
Mo 09.03.09 14:47:11.000	off	off	off	off	off	off	0	0	0	0	0	0	off	ON	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off
Mo 09.03.09 14:02:00.100	off	off	off	off	off	off	0	0	0	0	0	0	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off
Mo 09.03.09 14:00:00.100	off	off	off	off	off	off	0	0	0	0	0	0	off	off	off	off	off	off	off	off	off	off	off	off	off	ON	off	off	off	off
Mo 09.03.09 13:02:00.100	off	off	off	off	off	off	0	0	0	0	0	0	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off
Mo 09.03.09 13:00:00.100	off	off	off	off	off	off	0	0	0	0	0	0	off	off	off	off	off	off	off	off	off	off	off	off	off	ON	off	off	off	off
Mo 09.03.09 11:02:00.100	off	off	off	off	off	off	0	0	0	0	0	0	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off
Mo 09.03.09 11:00:00.100	off	off	off	off	off	off	0	0	0	0	0	0	off	off	off	off	off	off	off	off	off	off	off	off	off	ON	off	off	off	off
Mo 09.03.09 10:02:00.100	off	off	off	off	off	off	0	0	0	0	0	0	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off
Mo 09.03.09 10:00:00.100	off	off	off	off	off	off	0	0	0	0	0	0	off	off	off	off	off	off	off	off	off	off	off	off	off	ON	off	off	off	off
Mo 09.03.09 08:24:00.100	off	off	off	off	off	off	0	0	0	0	0	0	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off
Mo 09.03.09 08:22:00.100	off	off	off	off	off	off	0	0	0	0	0	0	off	off	off	off	off	off	off	off	off	off	off	off	off	ON	off	off	off	off
Fr 06.03.09 16:02:00.100	off	off	off	off	off	off	0	0	0	0	0	0	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off
Fr 06.03.09 16:00:00.100	off	off	off	off	off	off	0	0	0	0	0	0	off	off	off	off	off	off	off	off	off	off	off	off	off	ON	off	off	off	off
Fr 06.03.09 15:58:00.100	off	off	off	off	off	off	0	0	0	0	0	0	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off	off

<< >>

Browser view of the stored data

5.2.3 Labeling and texts

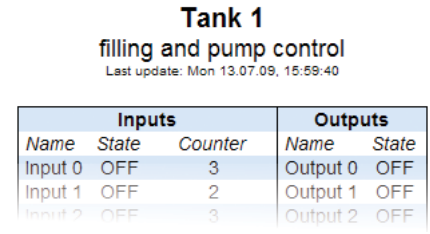
Required access rights: *Administrator*

The appearance of the Web-IO Digital Logger pages in the browser can be individually designed and changed within certain limits.

Device name and description
In the navigation tree under *Config >> Device >> Text* you can change device name and description of the Web-IO Digital Logger.

If the description takes up more than one line, insert the <br> tag where the line break needs to be made.

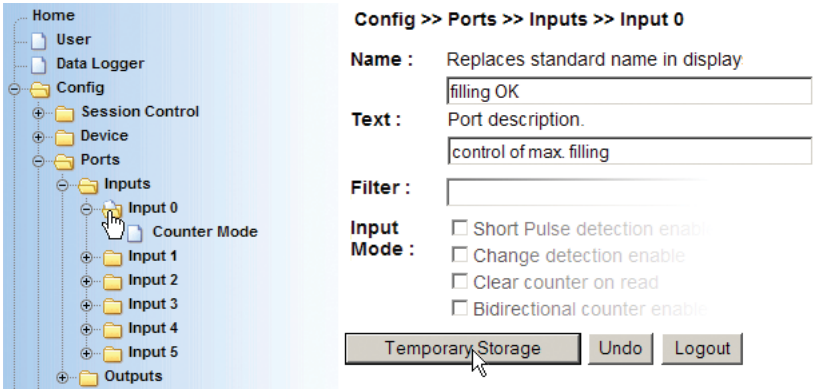
In the browser window you can adapt the display after sending and activating the settings.



Port name and description

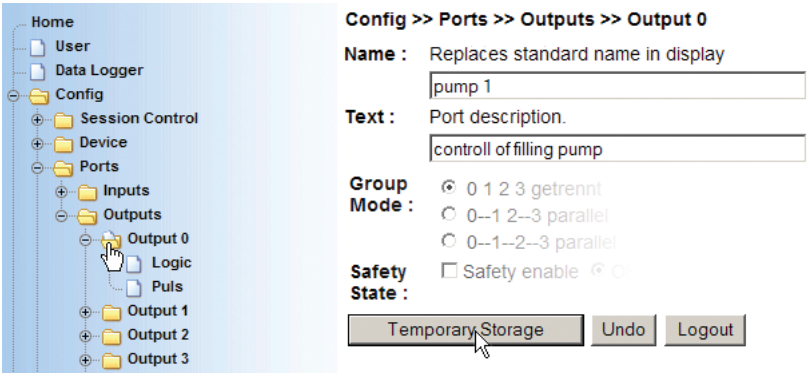
You can also set up individual names and descriptions for the inputs and outputs.

To change the input name, select in the navigation tree *Config >> Ports >> Inputs* and for Input 0 *>> Input 0*:



STOP *Please do not make any changes to the settings for* Input Mode *at this time. For detailed information about these parameters, refer to section* General Settings.

For changes to the output names select *config >> Ports >> Outputs* and for example *>> Output 0* for Output 0



Please do not make any changes to the settings for Group Mode *and* Safety State at this time. *For detailed information about these settings, refer to section* General Settings.

In the browser window you can adapt the display after sending and activating the settings.

Tank 1

filling and pump control

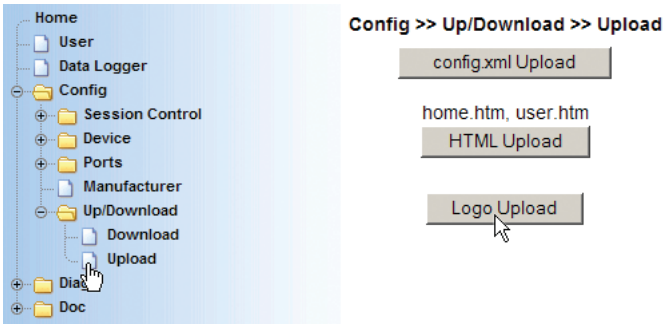
Last update: Mon 13.07.09, 16:18:42

Inputs			Outputs	
Name	*State*	*Counter*	*Name*	*State*
filling OK	OFF	3	pump 1	OFF
Input 1	OFF	2	Output 1	OFF
Input 2	OFF	3	Output 2	OFF

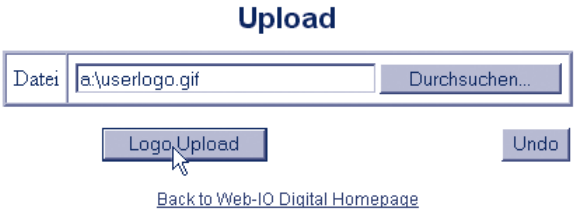
Clicking on the name of the inputs and outputs opens the associated description.

Logo in navigation frame

To enable a customized as possible appearance of the Web-IO Digital Logger start page, it is possible to replace the W&T logo in the navigation frame with any logo of your choice. Select in the navigation tree *Config >> Up/Download >> Upload*



Click on the button *logo upload*.

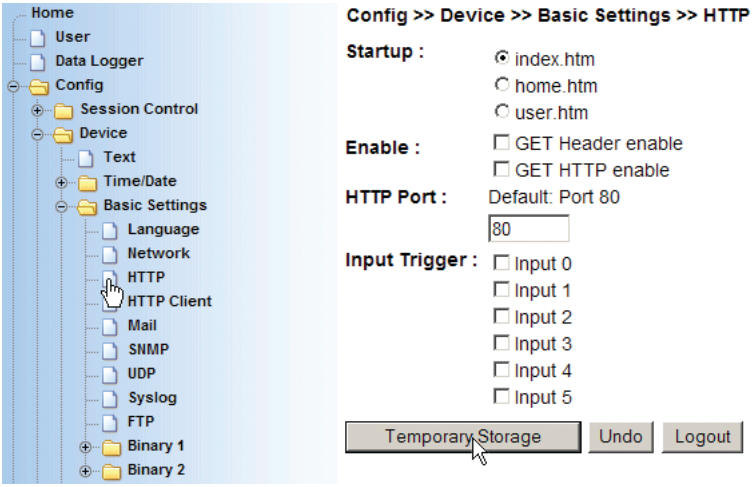


Select the desired JPG- or GIF file and click on the button *logo.gif upload.*

Note that the dimensions of the logo must agree approximately with those of the W&T logo (width max. 200 pixels, height max. 100 pixels) so that the navigation frame can be seen in its entirety.

Display at start
Select in the navigation tree *Config >> Basic Settings >> HTTP*

Required access rights: *Administrator*



Startup

Here you can specify how the Web-IO Digital Logger is displayed in the browser when opened.

- index.htm
  Both the navigation tree and the Homepage are displayed.
- home.htm
  Only the Homepage is shown, not the navigation tree.
- user.htm
  Only the user page is shown, without the navigation tree.

HTTP Port

Here you may enter a port for HTTP access other than the standard Port 80.

This setting should be changed only if there is a compelling reason for doing so. Please consult your network administrator.

If the HTTP port is changed, the Web-IO Digital Logger will be reachable from the browser only by additionally sp;ecifying the selected HTTP port.

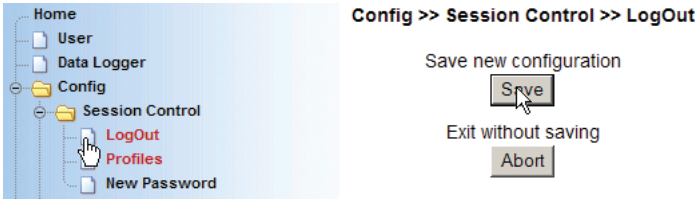
Example for HTTP Port = 85.

This applies as well to all configuration accesses.

The item GET Header *enable and the* Input Trigger *have no effect on the visualizing of the Web page and are explained in greater detail in the section „Socket programming with command strings".*

In the browser window you can adapt the display after sending and activating the settings.

After you have made all your settings, use the *Logout* button located on each configuration page or select >> Session Control >> LogOut in the Config menu.



After clicking on the *Save* button, all the Web-IO Digital Logger settings are updated and the start page is reopened in Default User Mode.

5.2.4 Creating your own Web site for the Web-IO Digital Logger

Required access rights: *Administrator*

The start or home page contains a clear summary of all the Web-IO Digital Logger functions available in the browser.

It is also possible to create another Web site designed more according to your own needs and to make it accessible in the Web-IO Digital Logger.

To incorporate the Web-IO Digital Logger-specific display objects and control elements, the Web-IO Digital Logger supports special HTML tags.

These tags consist of

```
<w&t_tags=
```

and the actual function invoking.

```
<w&t_tags=time>
```

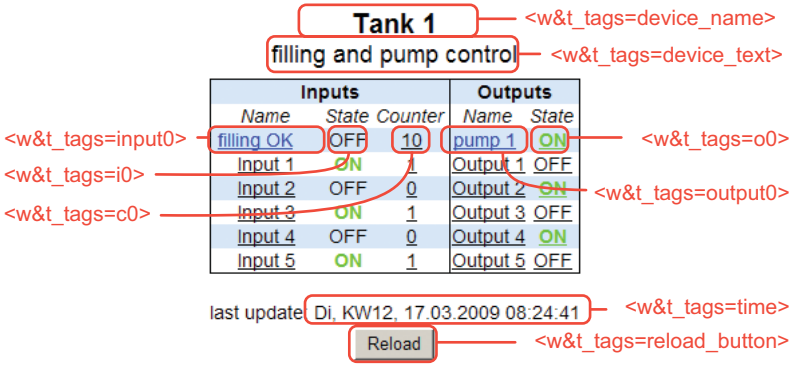
for example displays the current system time in addition to the date.

The following functions are available:

Invoke / Tag	Function
device_name	Inserts the name assigned for the Web-IO in the web site
device_text	Inserts the description defined for the Web-IO in the web site.
input**x**	Inserts the name specified for input no. **x**. **x can be a number between 0-11 and indicates which input the invoke refers to. This applies as well to the output and counter invokes described below.**
i**x**	Indicates the state (ON/OFF) of the input corresponding to **x**.
bci**x**	Represents the background of a table cell depending on the state of Input **x**. OFF = no background color, ON = green.
c**x**	Inserts the counter state of the counter for Input **x** in the web site.
output**x**	Inserts the name specified for Output **x**.
o**x**	Shows the state (ON/OFF) of the output corresponding to **x**. When logging in with Operator or Administrator rights, the state indication is given a hyperlink. Clicking on this link changes the state of the corresponding output and refreshes the web site.
bco**x**	Represents the background of a table cell depending on the state of Output **x**. OFF = no background color, ON = green.
access:xxxx:yyyy	Can be added on a link to the pages home.htm, user.htm or index.htm. When there is an Administrator login multiple outputs are switched at the same time. **xxxx** represents in hex format which outputs are permitted to change. **yyyy** represents in hex format to what state the outputs should be switched.
time	Inserts the system time and data of the Web-IO in the Web site.
reload_button	Inserts a button having a reload function in the Web site.
session	When logging in with Operator or Adminstrator rights, the Web IO generates a Session ID. When the page is opened again this Session ID must be specified as a parameter, so that for example when jumping back through a link the access rights are retained.

An example for use of W&T tags

In terms of structure the user page could look as follows:



The source text always begins with <user.htm>. This tag is used for page identification and is not sent to the browser when opening from the Web-IO Digital Logger.

```
<user.htm>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1250">
<title>Web-IO6x6DigitalLogger,User</title>
```

The appearance and color of the individual elements are defined in a CSS style.

```
<style type="text/css">
<!--
* { font-family:arial; }
a {color:#000000;}
table { font-size:14px; text-align:center;}
.borderLeft { border-left:1px solid #000000; }
.description { font-size:18px; text-align:center; }
.italic { font-style:italic; }
.name { font-size:20px; font-weight:bold; text-align:center }
.table { background-color:#d6e8ff; border-collapse:collapse;
border:1px solid #000000; }
.whiteBack { background-color:#ffffff; }
.IO_state_0 { color:#000000;}
.IO_state_0 a {color:#000000;}
```

```
.IO_state_1 { color:#66CC00; font-weight: bold;}
.IO_state_1 a {color:#66CC00;}
-->
</style>
</head>
```

Die W&T-Tags finden sich im Bodybereich der Webseite wieder

```
<body>
<div class="name"><w&t_tags=device_name></div>
<div class="description"><w&t_tags=device_text></div>
<table align="center" class="table">
<tr>
  <th colspan="3">Inputs</th>
  <th class="borderLeft" colspan="2">Outputs</th>
</tr>
<tr class="italic whiteBack">
  <td>Name</td>
  <td>State</td>
  <td>Counter</td>
  <td class="borderLeft sideSpace">Name</td>
  <td>State</td>
</tr>
<tr>
  <td><w&t_tags=input0></td>
  <td <w&t_tags=bci0>><w&t_tags=i0></td>
  <td><w&t_tags=c0></td>
  <td><w&t_tags=output0></td>
  <td <w&t_tags=bco0>><w&t_tags=o0></td>
</tr>
<tr class="whiteBack">
  <td><w&t_tags=input1></td>
  <td <w&t_tags=bci1>><w&t_tags=i1></td>
  <td><w&t_tags=c1></td>
  <td><w&t_tags=output1></td>
  <td <w&t_tags=bco1>><w&t_tags=o1></td>
</tr>
<tr>
  <td><w&t_tags=input2></td>
  <td <w&t_tags=bci2>><w&t_tags=i2></td>
```

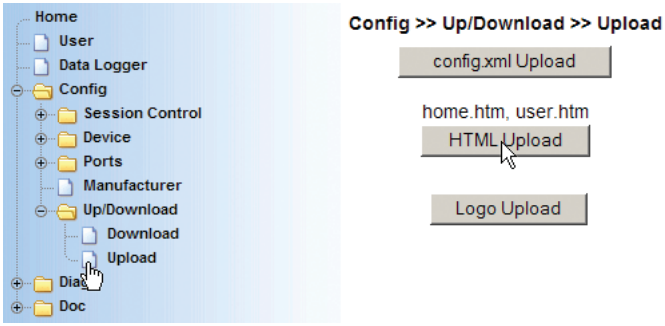
```
  <td><w&t_tags=c2></td>
  <td><w&t_tags=output2></td>
  <td <w&t_tags=bco2>><w&t_tags=o2></td>
</tr>
<tr class="whiteBack">
  <td><w&t_tags=input3></td>
  <td <w&t_tags=bci3>><w&t_tags=i3></td>
  <td><w&t_tags=c3></td>
  <td><w&t_tags=output3></td>
  <td <w&t_tags=bco3>><w&t_tags=o3></td>
</tr>
<tr>
  <td><w&t_tags=input4></td>
  <td <w&t_tags=bci4>><w&t_tags=i4></td>
  <td><w&t_tags=c4></td>
  <td><w&t_tags=output4></td>
  <td <w&t_tags=bco4>><w&t_tags=o4></td>
</tr>
<tr class="whiteBack">
  <td><w&t_tags=input5></td>
  <td <w&t_tags=bci5>><w&t_tags=i5></td>
  <td><w&t_tags=c5></td>
  <td><w&t_tags=output5></td>
  <td <w&t_tags=bco5>><w&t_tags=o5></td>
</tr>
</table>
<br>
<table  align="center">
<tr>
  <td colspan="5">last update: <w&t_tags=time></td>
</tr>
<tr>
  <td colspan="5"><form action="user.htm" method="GET" ><w&t_tags=reload_button>
   </form></td>
</tr>
</table>
</body>
</html>
```

The tags <w&t_tags=bci0> and <w&t_tags=bco0> play a special role. They are replaced by CSS style classes when the

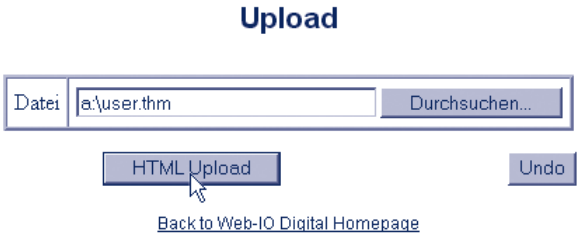
page is opened by the Web-IO Digital Logger. This means the color reproduction of the input and output states can be individually determined by status.

After the user Web page has been created, it must still be uploaded to the Web-IO Digital Logger.

Select in the navigation tree *Config >> Up/Download >> Upload*



and click on *HTML-Upload*



Once the upload is completed, the new settings still must be saved.

In the Config menu select >> Session Control >> LogOut.

After clicking on the Save button, the Web-IO Digital Logger is restarted and the newly created Web site can be opened.

Links and back links

Some applications may make it necessary to link from the user or home page of the Web-IO Digital Logger to a Web site which is located on another HTTP server.

If a back link then navigates you back to a Web-IO page, this is done using standard user access rights, even if before there was an Administrator or Operator login.

To enable a back link with the existing access rights, there is the <w&t_tags=session> tag. This tag is a placeholder for a session ID.

The corresponding link would then have to be inserted in the page as follows:

```
<ahref="http://wut.hs2/webioback.htm<w&t_tags=session>"target="_top">Testlink</a>
```

When the link is opened, the Web-IO Digital Logger replaces the session tag with the session ID and passes it to the newly opened Web site.

To use the session ID for a back link, a small Javascript must be inserted into the external page which accepts the session ID when opening the page and inserts it in the back link.

```
<html><head>

<script language="JavaScript">
session=window.location.search

function setbacklink() {
 var backlinkurl = "http://172.16.232.17/index.htm";
 var backlink = backlinkurl.concat(session);
 window.location.href = backlink;
}
</script>
```

In the body area of the page the function *setbacklink* can be used as a hyperlink.

```
</head><body>
A test page is now opened with the passed session. <a href="javascript:setbacklink();"
>Open
test page</a>
</body></html>
```

This ensures that the access authorization remains intact after jumping back to the Web-IO Digital Logger.

5.2.5 A user page using AJAX

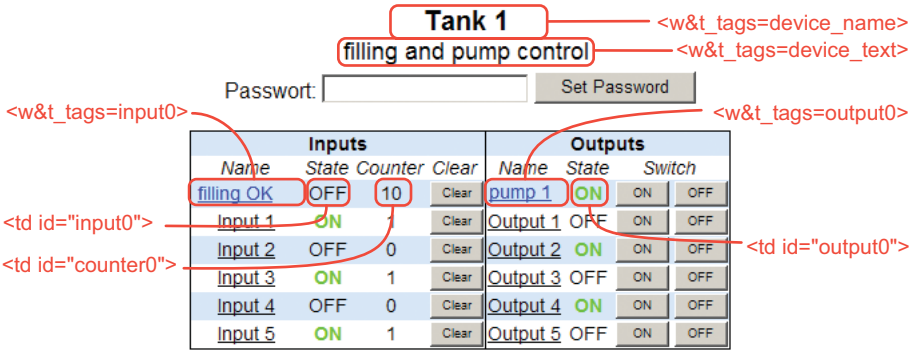
One significant disadvantage of the pure W&T tag Web page is the absence of dynamics. The displayed states are not automatically adjusted when a state on the Web-IO Digital Logger changes.

This deficiency can be overcome with a little JavaScript and the HTTP Request Object of the browser.

The HTML objects which need to be changed dynamically are given an ID in the HTML source text. JavaScript can use this ID to still change the properties of the HTML object even after loading the Web page.

Static content, such as the names of inputs and outputs, can also be inserted using W&T tags.

An AJAX user page could look like this:



The HTML source text begins with <user.htm>. This tag is used for page identification and is not sent by the Web-IO Digital Logger to the browser when the page is opened.

```
<user.htm>
```

Danach beginnt der eigentliche HTML-Kopf.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1250">
<title>Web-IO6x6DigitalLogger,User</title>
```

The appearance and color of the individual elements are defined in a CSS style.

```
<style type="text/css">
<!--
* { font-family:arial; }
a {color:#000000;}
table { font-size:14px; text-align:center;}
.borderLeft { border-left:1px solid #000000; }
.button { font-size:9px; width:40px; }
.description { font-size:18px; text-align:center; }
.italic { font-style:italic; }
.name { font-size:20px; font-weight:bold; text-align:center }
.table { background-color:#d6e8ff; border-collapse:collapse; border:1px solid #000000; }
```

```
.whiteBack { background-color:#ffffff; }
-->
</style>
<script language="JavaScript" type="text/javascript">
<!--
```

Here is where the JavaScript functions are located, which we will get to later. **JavaScript function calls** are shown within the HTML section in green for the purpose of clarity!

```
//-->
</script>
</head>
<body>
<div class="name"><w&t_tags=device_name></div>
<div class="description"><w&t_tags=device_text></div>
<br>
    <form>
        <div align="center">Passwort:
                  <input type="password" name="tfPassword" maxlength="31" size="20">
                   <input type="button" value="Set Password" onclick="setPassword()">
        </div>
    </form>
<table align="center" class="table">
<tr>
  <th colspan="4">Inputs</th>
  <th class="borderLeft" colspan="3">Outputs</th>
</tr>
<tr class="italic whiteBack">
  <td>Name</td>
  <td>State</td>
  <td>Counter</td>
  <td>Clear</td>
  <td class="borderLeft">Name</td>
  <td>State</td>
  <td>Switch</td>
</tr>
<tr>
  <td><w&t_tags=input0></td>
  <td id="input0">-</td>
  <td id="counter0">-</td>
```

```
  <td>
    <input class="button" onclick="clearCounter(0);" type="button" value="Clear">
  </td>
  <td class="borderLeft"><w&t_tags=output0></td>
  <td id="output0">-</td>
  <td >
    <input class="button" onclick="setOutput(0);" type="button" value="ON"> 
    <input class="button" onclick="resetOutput(0);" type="button" value="OFF">
  </td>
</tr>
<tr class="whiteBack">
  <td><w&t_tags=input1></td>
  <td id="input1">-</td>
  <td id="counter1">-</td>
  <td>
    <input class="button" onclick="clearCounter(1);" type="button" value="Clear">
  </td>
  <td class="borderLeft"><w&t_tags=output1></td>
  <td id="output1">-</td>
  <td>
    <input class="button" onclick="setOutput(1);" type="button" value="ON"> 
    <input class="button" onclick="resetOutput(1);" type="button" value="OFF">
  </td>
</tr>
<tr>
  <td><w&t_tags=input2></td>
  <td id="input2">-</td>
  <td id="counter2">-</td>
  <td>
    <input class="button" onclick="clearCounter(2);" type="button" value="Clear">
  </td>
  <td class="borderLeft"><w&t_tags=output2></td>
  <td id="output2">-</td>
  <td>
    <input class="button" onclick="setOutput(2);" type="button" value="ON"> 
    <input class="button" onclick="resetOutput(2);" type="button" value="OFF">
  </td>
</tr>
<tr class="whiteBack">
  <td><w&t_tags=input3></td>
  <td id="input3">-</td>
```

```
  <td id="counter3">-</td>
  <td>
    <input class="button" onclick="clearCounter(3);" type="button" value="Clear">
  </td>
  <td class="borderLeft"><w&t_tags=output3></td>
  <td id="output3">-</td>
  <td>
    <input class="button" onclick="setOutput(3);" type="button" value="ON"> 
    <input class="button" onclick="resetOutput(3);" type="button" value="OFF">
  </td>
</tr>
<tr>
  <td ><w&t_tags=input4></td>
  <td id="input4">-</td>
  <td id="counter4">-</td>
  <td>
    <input class="button" onclick="clearCounter(4);" type="button" value="Clear">
  </td>
  <td class="borderLeft"><w&t_tags=output4></td>
  <td id="output4">-</td>
  <td>
    <input class="button" onclick="setOutput(4);" type="button" value="ON"> 
    <input class="button" onclick="resetOutput(4);" type="button" value="OFF">
  </td>
</tr>
<tr class="whiteBack">
  <td><w&t_tags=input5></td>
  <td id="input5">-</td>
  <td id="counter5">-</td>
  <td>
    <input class="button" onclick="clearCounter(5);" type="button" value="Clear">
  </td>
  <td class="borderLeft"><w&t_tags=output5></td>
  <td id="output5">-</td>
  <td >
    <input class="button" onclick="setOutput(5);" type="button" value="ON"> 
    <input class="button" onclick="resetOutput(5);" type="button" value="OFF">
  </td>
</tr>
</table>
</body>
```

```
</html>
```

The HTML section of the Web page forms the basic structure for dynamic representation. JavaScript puts life in this structure.

First the declaration of some global variables in the JavaScript section is necessary.

```
var maxio = 6;                  //Anzahl der IOs
var setoutputrequest = maxio;  //dient zur Signalisierung welcher Output (0-5)
                               //veränder werden soll. Liegt keine Anforderung vor
                            //wird setoutputrequest = maxio gesetzt.
var clearrequest = maxio;      //dient zur Signalisierung welcher Counter gelöscht
                            //werden soll (Vorgehen wie beisetoutputrequest )
var applicationstep = 0;       //Das Script arbeitet in drei Schritten
                                   //0 = Inputs, 1 = Outputs, 3 = Counter lesen
var interval = 500;                //Ziet in ms zwischen den Schritten
var iopassword = '';               //speichert das Passwort zwischen
outputvalue = new Array(maxio); //gewünschter Outputstatus (true=ON oder False=OFF
for (i=0; i<maxio; i++)
{   outputvalue[i] = false;
}
```

The Web-IO Digital Logger returns the status of inputs and outputs as a hexadecimal string. The following function converts this into a decimal number.

```
function HexToInt(HexStr)
{ var TempVal;
  var HexVal=0;
  for( var i=0; i<HexStr.length;i++)
    { if (HexStr.charCodeAt(i) > 57)
      { TempVal = HexStr.charCodeAt(i) - 55;
        }
        else
        { TempVal = HexStr.charCodeAt(i) - 48;
        }
        HexVal=HexVal+TempVal*Math.pow(16, HexStr.length-i-1);
    }
    return HexVal;
}
```

Before JavaScript starts its work, the user must invoke the setPassword function.

```
function setPassword()
{   iopassword = document.forms[0].tfPassword.value;
    document.forms[0].tfPassword.value = '';
    maintimer = setTimeout('CommandLoop()', 500);
}
```

This function accepts the password, which is later inclouded in the command strings and starts the timer which is resonsible for continuous running of the program steps.

The key to the dynamics is cyclical communication with the Web-IO Digital Logger. Here JavaScript invokes an instance of the XMLHTTPRequest object, sets the required HTTP header options and sends the desired command to the Web-IO Digital Logger and accepts the reply. This task is handled by the DataRequest function.

```
function DataRequest(sendstring)
{   try
  { // Internet Explorer
    if( window.ActiveXObject )
    {   var xmlHttp = new ActiveXObject( "Microsoft.XMLHTTP" );
    }
    // Mozilla, Opera und Safari
        else if( window.XMLHttpRequest )
        {   var xmlHttp = new XMLHttpRequest();
        }
    }
    // loading of xmlhttp object failed
```

```
    catch( excNotLoadable )
    {   xmlHttp = false;
    }
    if (xmlHttp)
    {  xmlHttp.onreadystatechange = function()
       { var ReceiveStr;
          if (xmlHttp.readyState == 4)
          {   if (xmlHttp.status == 200)
             {ReceiveStr = xmlHttp.responseText;
               DisplayUpdate(ReceiveStr);
             }
             xmlHttp=null;
       }
     }
     xmlHttp.open("GET", sendstring, true);
        xmlHttp.setRequestHeader("Keep-Alive","25");
     xmlHttp.setRequestHeader("Connection","close");
     xmlHttp.setRequestHeader("Cache-Control", "no-store, no-cache, must-revalidate");
        xmlHttp.setRequestHeader("Expires", "Sat, 05 Nov 2005 00:00:00 GMT");
        xmlHttp.setRequestHeader("Pragma","no-cache");
        xmlHttp.send(null);
    }
}
```

Which command should be sent is determined by the Commandloop function, which invokes itself cyclically using a timer inverval. A complete list of all possible commands is found on page 110. The function first checks whether a request for setting an output or clearing a counter is present. If not, the inputs, outputs or counters are read depending on the application step.

```
function CommandLoop()
{ var commandstring ='';
  if (setoutputrequest<maxio)
  { if (outputvalue[setoutputrequest]==true)
        {   commandstring = 'outputaccess'+setoutputrequest+'?PW='+iopassword+'&State=ON&';
        }
        else
        {   commandstring = 'outputaccess'+setoutputrequest+'?PW='+iopassword+'&State=OFF&';
        }
```

```
        setoutputrequest = maxio;
    }
    else
    {   if (clearrequest<maxio)
        {   commandstring = 'counterclear'+clearrequest+'?PW='+iopassword+'&';
            clearrequest=maxio;
        }
        else
        {   switch(applicationstep)
          { case 0:
            commandstring = 'input?PW=' + iopassword + '&';
            break;
            case 1:
              commandstring = 'output?PW=' + iopassword + '&';
            break;
            case 2:
              commandstring = 'counter?PW=' + iopassword + '&';
            break;
        }
      applicationstep++;
      if(applicationstep>2)
        {   applicationstep=0;
        }
        }
    }
    DataRequest(commandstring);
    maintimer = setTimeout("CommandLoop()", interval);
}
```

Evaluation of the replies and refreshing of the browser display is handled by the DisplayUpdate function.

In the document.getElementById method the content to be changed in the HTML basic structure is identified by the ID and rewritten according to the current state.

```
function DisplayUpdate(ReceiveStr)
{ var HexVal;
  var state = false;
  // Display Input state
```

```
if (ReceiveStr.substring(0,5)=='input')
{ HexVal=HexToInt(ReceiveStr.substring(6,10));
  for (var i=0;i<maxio;i++)
  {   if ((HexVal & Math.pow(2,i)) == Math.pow(2,i))
    {  state = true;
    }
    else
    {  state = false;
         }
         document.getElementById('input'+i).firstChild.data =
         ( !state ) ? 'OFF' : 'ON';
         document.getElementById('input'+i).style.color=
         ( !state ) ? '#000000' : '#66CC00';
         document.getElementById('input'+i).style.fontWeight=
         ( !state ) ? 'normal' : 'bold';
      }
  }
  // Display Output state
if (ReceiveStr.substring(0,6)=='output')
  {   HexVal=HexToInt(ReceiveStr.substring(7,11));
    for (var i=0;i<maxio;i++)
    {  if ((HexVal & Math.pow(2,i)) == Math.pow(2,i))
         {   state = true;
         }
         else
         {   state = false;
         }
         document.getElementById('output'+i).innerHTML =
           ( !state) ? 'OFF' : 'ON';
         document.getElementById('output'+i).style.color=
           ( !state ) ? '#000000' : '#66CC00';
         document.getElementById('output'+i).style.fontWeight=
           ( !state ) ? 'normal' : 'bold';
      }
  }
//Display Counter
if (ReceiveStr.substring(0,7)=='counter')
{ if (ReceiveStr.substring(7,8)==';')
  {   var countervalue=ReceiveStr.split(';');
    for (var i=0;i<maxio;i++)
    { document.getElementById('counter'+i).innerHTML = countervalue[i+1];
    }
```

```
    }
      else
      { if (ReceiveStr.substring(9,10)==';')
        {   i=(ReceiveStr.substring(7,9));
          document.getElementById('counter'+i).innerHTML =
               ReceiveStr.substring(10,slength);
        }
        else
        {   i=(ReceiveStr.substring(7,8));
          document.getElementById('counter'+i).innerHTML =
               ReceiveStr.substring(9,slength);
        }
      }
    }
}
```

The following functions are responsible for setting/resetting the outputs or clearing the counters:

```
function setOutput(iNr)
{   outputvalue[iNr]=true;
  setoutputrequest=iNr;
}

function resetOutput(iNr)
{   outputvalue[iNr]=false;
  setoutputrequest=iNr;
}

function clearCounter(iNr)
{   clearrequest = iNr;
}
```

Invoking these functions is done from the HTML basic structure using the onClick events of the buttons.

Based on this example any desired user pages can be created.

5.2.6 Java Applets - Auto-updating in the browser

Visualization of states in the browser, based on pure HTTP or HTML, permits only a snapshot. Displayed is the status at the time the Web page was opened. A status change can only be displayed by reloading the complete Web page.

To allow current states to be displayed in the browser, a Java applett is implemented in the Web-IO 12xDigital.

In order to permit the most flexible utilization, the Java applet opens pre-defined JavaScript functions which must be declared in the header of the HTML page.

In addition the Java applet provides functions which allow you to set outputs and clear counters using JavaScript.

Working with the Java applet

The applet is incorporated using the <applet> HTML tag, where the standard parameters *archive* and *code* are set as follows:

```
archive="dio.jar"
code="dio.class"
```

The remaining parameters are set individually according to the application.

```
name="dio0"
```

The designator *dio* was chosen randomly here and can be replaced with any desired. The only stipulation is that the same name is used when invoking the applet functions with JavaScript.

```
codebase="IP-AddressoftheWeb-IODigitalLogger"
```

If the Web page from which the applet is loaded is loaded directly from the Web-IO Digital Logger as *user.htm* or *home.htm*, the *codebase* parameter is not needed.

```
width="0"      Specified in pixels
height="0"     Specifiedinpixels
```

Since the applet does not contain a visual component, *width* and *height* should always be 0.

In addition you can set several more Web-IO Digital Logger-specific parameters which are inserted in a separate <param> tag:

device
Whole-number value between 0 and 255. The *device* parameter is used for indexing when more than one Web-IO Digital Logger needs to be operated on a Web site.

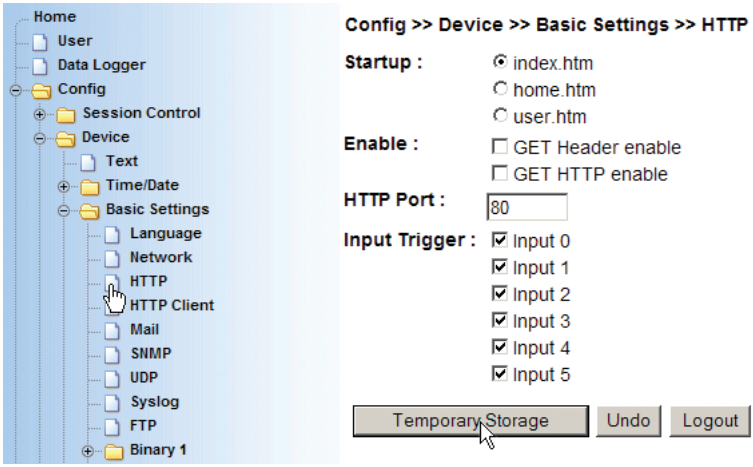
If working with just one Web-IO Digital Logger, set the value to 0. For all additional Web-IO Digital Loggers being incorporated into the Web site, the value is raised by 1.

showerrors
If errors occur when using the Java applet, this parameter can be used to specify whether an error message in the form of an alert window should be generated.

inputpolling
This parameter is used to specify when the applet is started whether the inputs should be checked for their state at the interval set under *pollingrate*.

Als *value* kann für *inputpolling on* oder *off* gesetzt werden.

In order to reduce the data load on the network, you can use the navigation tree of the Web-IO Digital Logger under *Config >> Device >> Basic Settings >> HTTP* to set the *Input Trigger* for the monitored inputs.



In this case *value* can be set to *off*. This means there is a data load for updating the inputs when an input changes.

For the changes to the input trigger to become effective, the new settings must be saved under *Control >> Logout*.

outputpolling
This parameter is used to specify when starting the applet whether the outputs should be checked for their state at the interval set under *pollingrate*.

The *value* for *outputpolling* can be set to *on* or *off*.

counterpolling
This parameter is used to specify when starting the applet whether the counters should be checked for their state at the interval set under *pollingrate*.

The *value* for *counterpolling* can be set to *on* or *off*.

pollingrate
Mit diesem Parameter wird durch value festgelegt, in welchem Intevall (in ms) Inputs, Outputs oder Counter aktualisiert werden sollen.

Here is an example:

```
<applet name="dio0" archive="dio.jar" code="dio.class"
      codebase="http://172.16.232.11" width="0" height="0">
  <param name="device" value="0">
  <param name="showerrors" value="off">
  <param name="inputpolling" value="on">
  <param name="outputpolling" value="on">
  <param name="counterpolling" value="on">
  <param name="pollingrate" value="2000">
</applet>
```

Both the parameters and the values are case-sensitive.

5.2.7 Using the applet with JavaScript

To work with the applet, corresponding JavaScript functions must be declared in the header of the Web page.

There are 4 functions for read-only access:

Password protection

```
<script language="JavaScript" type="text/javascript">
<!--
function setPassword()
  { document.applets["dio0"].setPassword('Administratorpasswort');}
```

If an Administrator or Operator password has been set for the Web-IO Digital Logger, the *setpassword* function must be invoked so that you can work with the applet. Instead of a static password, the password may of course also be queried by the user using an input field and then passed on.

In any case, setPassword can only be invoked once the Java applet has been started, since otherwise runtime errors would occur.

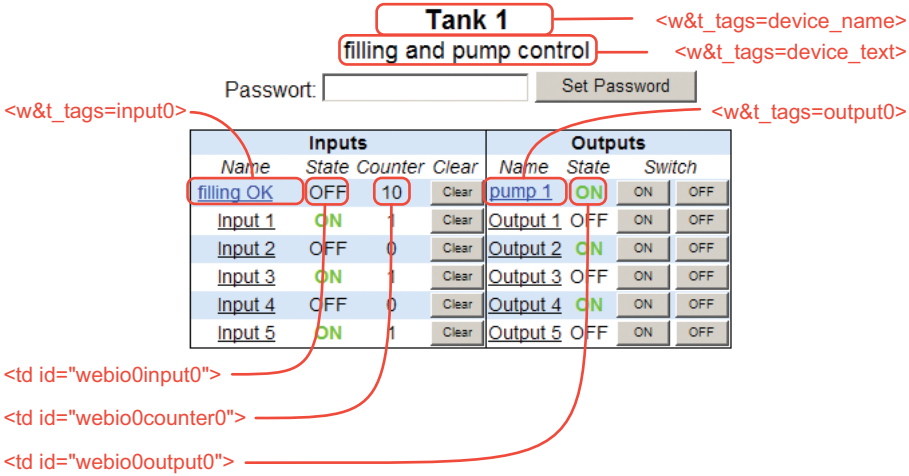
An example in which the password is passed on using an input field can be found by opening the Web page app.htm *in the Web-IO Digital Logger.*

Read functions

```
function inputChanged( iDevice, iNr, iVal )
  { Program code executed when there is a change on the inputs}
function outputChanged( iDevice, iNr, iVal )
  { Program code executed when there is a change on the outputs}
function counterChanged( iDevice, iNr, iVal )
  { Programcodeexecutedwhenthereisachangeinthecounters}
```

The three functions above are invoked by the applet when there is a change on the inputs or outputs or in the counters. *iDevice* shows on which device a change has happened. *iNr* specifies which input, output or counter has changed. The variable *iVal* conveys the current state.

Please note that the names of the functions are case-sensitive.



The HTML source text begins with <user.htm>. This tag is used for page identification and is not sent to the browser when opened.

```
<user.htm>
```

Then the actual HTML head begins.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1250">
<title>Web-IO6x6DigitalLogger,User</title>
```

The appearance and color of the individual elements are defined in a CSS style.

```
<style type="text/css">
<!--
* { font-family:arial; }
a {color:#000000;}
table { font-size:14px; text-align:center;}
.borderLeft { border-left:1px solid #000000; }
.button { font-size:9px; width:40px; }
.description { font-size:18px; text-align:center; }
.italic { font-style:italic; }
.name { font-size:20px; font-weight:bold; text-align:center }
.table { background-color:#d6e8ff; border-collapse:collapse; border:1px solid #000000; }
.whiteBack { background-color:#ffffff; }
-->
</style>
<script language="JavaScript" type="text/javascript">
<!--
```

Here is where the JavaScript functions are located, which we will get to later. **JavaScript function calls** are shown within the HTML section in green for the purpose of clarity!

```
//-->
</script>
</head>
```

```
<body>
```

First the applet is incorporated.

```
<applet name="dio0" archive="dio.jar" code="dio.class"
 width="0" height="0" mayscript>
    <param name="device" value="0">
    <param name="showerrors" value="off">
    <param name="inputpolling" value="on">
    <param name="outputpolling" value="on">
    <param name="counterpolling" value="on">
    <param name="pollingrate" value="1500">
    Java ist nicht aktiviert oder wird nicht unterst&uuml;tzt
</applet>
<div class="name"><w&t_tags=device_name></div>
<div class="description"><w&t_tags=device_text></div>
<br>
    <form>
        <div align="center">Passwort:
                  <input type="password" name="tfPassword" maxlength="31" size="20">
                   <input type="button" value="Set Password" onclick="setPassword(0)">
        </div>
    </form>
<table align="center" class="table">
<tr>
  <th colspan="4">Inputs</th>
  <th class="borderLeft" colspan="3">Outputs</th>
</tr>
<tr class="italic whiteBack">
  <td>Name</td>
  <td>State</td>
  <td>Counter</td>
  <td>Clear</td>
  <td class="borderLeft">Name</td>
  <td>State</td>
  <td>Switch</td>
</tr>
<tr>
  <td><w&t_tags=input0></td>
  <td id="webio0input0">-</td>
  <td id="webio0counter0">-</td>
```

```
  <td>
    <input class="button" onclick="clearCounter(0,0);" type="button" value="Clear">
  </td>
  <td class="borderLeft"><w&t_tags=output0></td>
  <td id="webio0output0">-</td>
  <td>
    <input class="button" onclick="setOutput(0,0);"
     type="button" value="ON"> 
    <input class="button" onclick="resetOutput(0,0);"
       type="button" value="OFF">
  </td>
</tr>
<tr class="whiteBack">
  <td><w&t_tags=input1></td>
  <td id="webio0input1">-</td>
  <td id="webio0counter1">-</td>
  <td>
    <input class="button" onclick="clearCounter(0,1);" type="button" value="Clear">
  </td>
  <td class="borderLeft"><w&t_tags=output1></td>
  <td id="webio0output1">-</td>
  <td>
    <input class="button" onclick="setOutput(0,1);"
       type="button" value="ON"> 
    <input class="button" onclick="resetOutput(0,1);"
       type="button" value="OFF">
  </td>
</tr>
<tr>
  <td><w&t_tags=input2></td>
  <td id="webio0input2">-</td>
  <td id="webio0counter2">-</td>
  <td>
    <input class="button" onclick="clearCounter(0,2);" type="button" value="Clear">
  </td>
  <td class="borderLeft"><w&t_tags=output2></td>
```

```
  <td id="webio0output2">-</td>
  <td>
    <input class="button" onclick="setOutput(0,2);"
       type="button" value="ON"> 
    <input class="button" onclick="resetOutput(0,2);"
       type="button" value="OFF">
  </td>
</tr>
<tr class="whiteBack">
  <td><w&t_tags=input3></td>
  <td id="webio0input3">-</td>
  <td id="webio0counter3">-</td>
  <td>
    <input class="button" onclick="clearCounter(0,3);" type="button" value="Clear">
  </td>
  <td class="borderLeft"><w&t_tags=output3></td>
  <td id="webio0output3">-</td>
  <td>
    <input class="button" onclick="setOutput(0,3);" type="button" value="ON"> 
    <input class="button" onclick="resetOutput(0,3);" type="button" value="OFF">
  </td>
</tr>
<tr>
  <td><w&t_tags=input4></td>
  <td id="webio0input4">-</td>
  <td id="webio0counter4">-</td>
  <td>
    <input class="button" onclick="clearCounter(0,4);" type="button" value="Clear">
  </td>
  <td class="borderLeft"><w&t_tags=output4></td>
  <td id="webio0output4">-</td>
  <td>
    <input class="button" onclick="setOutput(0,4);"
       type="button" value="ON"> 
    <input class="button" onclick="resetOutput(0,4);"
       type="button" value="OFF">
  </td>
</tr>
<tr class="whiteBack">
  <td><w&t_tags=input5></td>
  <td id="webio0input5">-</td>
```

```
  <td id="webio0counter5">-</td>
  <td>
    <input class="button" onclick="clearCounter(0,5);" type="button" value="Clear">
  </td>
  <td class="borderLeft"><w&t_tags=output5></td>
  <td id="webio0output5">-</td>
  <td>
    <input class="button" onclick="setOutput(0,5);"
       type="button" value="ON"> 
    <input class="button" onclick="resetOutput(0,5);"
       type="button" value="OFF">
  </td>
</tr>
</table>
</body>
</html>
```

The HTML page of the Web site forms the basic structure for dynamic display. JavaScript fills this framework with life. The following JavaScript functions are directly involved for this by the Java applet, whereby in addition to the Device Number and IP Port Number the current IO states and counter values are sent.

```
function inputChanged( iDevice, iNr, iVal )
{   document.getElementById( 'webio' + iDevice +'input' + iNr ).firstChild.data =
    ( !iVal ) ? 'OFF' : 'ON';
    document.getElementById( 'webio' + iDevice +'input' + iNr ).style.color=
    ( !iVal ) ? '#000000' : '#66CC00';
    document.getElementById( 'webio' + iDevice +'input' + iNr ).style.fontWeight=
    ( !iVal ) ? 'normal' : 'bold';
}

function outputChanged( iDevice, iNr, iVal )
{   document.getElementById( 'webio' + iDevice +'output' + iNr ).firstChild.data = ( !iVal )
? 'OFF' : 'ON';
    document.getElementById( 'webio' + iDevice +'output' + iNr ).style.color= ( !iVal ) ?
'#000000' : '#66CC00';
    document.getElementById( 'webio' + iDevice +'output' + iNr ).style.fontWeight= ( !iVal )
? 'normal' : 'bold';
}
```

```
function counterChanged( iDevice, iNr, iVal )
{   document.getElementById( 'webio' + iDevice  + 'counter' + iNr ).firstChild.data = iVal;
}
```

The user can employ corresponding buttons to set the outputs and clear the counters. The onClick methods for the buttons are used to invoke the following functionsm, which evaluate which action should occur and invoke the corresponding request in the Java applet.

```
function setOutput( iDevice, iOutput, iValue )
{   iOut = 0;
    iOut |= Math.pow( 2, iOutput );
    document.applets["dio"+iDevice].outputAccess( iOut, iValue );
}

function resetOutput( iDevice, iOutput, iValue )
{   iOut = 0;
    iOut |= Math.pow( 2, iOutput );
    document.applets["dio"+iDevice].outputAccess( iOut, iValue );
}

function clearCounter( iDevice, iValue )
{   iVal = 0;
    iVal |= Math.pow( 2, iValue );
    document.applets["dio"+iDevice].counterClear( iVal );
}

function setPassword(iDevice)
{   document.applets["dio0"].setPassword( document.forms[0].tfPassword.value );
    document.forms[0].tfPassword.value = '';
}
```

The following applies to processing inputs, outputs and counters: If the corresponding polling is enabled, the associated JavaScript function must be declared in the HTML text so that there are no runtime errors.

If you want the password to be automatically sent when opening the Web page, the setPassword function must

be invoked behind the applet tag.
In this case the password is stored unencrypted in the source text. For security reasons we therefore recommend using an input field for prompting the password.

Write functions

The applet provides functions for individual access to the outputs as well.

```
function setOutput( iDevice, iOutput)
        {   iOut = 0;
            iOut |= Math.pow( 2, iOutput );
            document.applets["dio"+iDevice].outputAccess( iOut, 0x0FFF);
        }
```

If you want to use JavaScript for accessing the outputs, the above function should be placed in the Web page head. iDevice indicates for which Web-IO Digital Logger an output should be switched. *iOutput* indicates which output should be set.

Within the function the JavaScript command *document.applets["dio"*+iDevice*].outputAccess(iOut, 0x0FFF)* is used to set the selected output to ON. *document.applets["dio"*+iDevice*].outputAccess(iOut, 0x0000)* on the other hand sets the selected output to OFF. If you want to switch more than one output at a time, you can use this command in separate functions. Thew variable *iOut* is a whole-number 16-bit value. By using mathematical functions for example or transmitting constants, the bits which are to be changed can be set to 1. In *iValue* you specify for the individual bits whether the corresponding output should be 0 (=OFF) or 1 (=ON).

Example in hexadecimal:

iOut = 0x0013, iValue = 0x0011

In this case outputs 0 and 4 are set to ON and output 1 to 0. All other outputs remain unchanged.

There is another function for clearing the counters.

```
function clearCounter( iDevice, iValue )
        {   iVal = 0;
            iVal |= Math.pow( 2, iValue );
            document.applets["dio"+iDevice].counterClear( iVal );
        }
```

When invoking this function *iValue* is used to indicate which of the counters should be cleared.

5.2.8 Limits when using the Java applet

When the Web-IO Digital Logger is accessed using the Java applet, there are differences in how this is done depending on the browser.

All Mozilla-based browsers as well as Opera do not allow write accesses to the Web-IO Digital Logger using the applet if the corresponding Web page for example is loaded by the local hard drive of the PC or by a Web server. If no password is assigned on the Web-IO Digital Logger, read access can at least be used.

Microsoft Internet Exlorer on the other hand can be used without limitations.

To be able to work with the Web-IO Digital Logger without restrictions, we recommend saving the Web page as home.htm or user.htm in the Web-IO Digital Logger. In this case the newest releases of most browsers will allow unrestricted access.

The number of users who can use a Web page working the the Java applet is limited to 3 at a time (if other access types than HTTP are used, for example BINARY or UDP-ASCII, the number of possible users is reduced).

A comprehensive example for using the Java applet can be found on the Web page app.htm in the Web-IO Digital Logger.

Open: http://<ip-adresse>/app.htm

5.3 Controlling the Logger with TCP/UDP Sockets

To access the Web-IO Digital Logger directly from your own applications, two basic methods are provided for using TCP or UDP sockets to read inputs and counters and set outputs.

Access via:

- command strings **ASCII**
- binary structures **BINARY**

5.3.1 Selecting the appropriate access mode

Command strings ASCII
In this mode the Web-IO Digital Logger works as a TCP server or as a UDP peer.

Inputs and counters can be read and outputs set by exchanging simple command strings.

Inputs are generally read by using polling. Event-controlled processing is only conditionally possible for imput patters defined in alarms.

Advantages:

- Since all higher level languages such as Visual Basic, Visual Basic for Applications, Delphi, C, C++, etc. allow uncomplicated handling of string variables, even inexperienced programmers will have little difficulty achieving the desired results.
- Under TCP Port 80 is used (otherwise HTTP), which is not blocked by most firewalls. This means that mostly no changes to the netework infrastructure are necessary.
- Using string variables together with TCP Port 80 makes it quite easy to integrate communication with the Web-IO Digital Logger into PHP and CGI scripts. The Web-IO Digital Logger can also be accessed from PLCs and other microcontrollers as long as there is a network connection.

Disadvantages:

- Only partial event-controlled programming possible. This limits its usefulness for time-critical applications.
- Only client applications possible, no server applications.

Summary: Access using command strings means the shortest development time for client applications in which processing speed is not the highest priority.

Binary structures BINARY
In this mode the Web-IO Digital Logger can function as a TCP client, TCP server or UDP peer. Access can be password-protected.

The Web-IO Digital Logger defines binary structures for the various functions such as reading inputs, setting outputs, etc. Access takes place solely by exchanging these structures.

Advantages:

- Flexible in its application, since client and server applications are possible
- Fast communication through use of structures. The status values are always at a defined location, and no strings need to be analyzed for their contents
- Event-triggered communication allows the application to respond immediately to changes in the inputs
- Password protected access makes a high degree of application security possible

Disadvantages:

- In higher languages which do not support binary structures a workaround in the form of binary strings or byte arrays must be found

Summary: BINARY mode should be used for professional applications which require a high level of processing speed.

Special forms of BINARY mode

- Box-to-Box
  Two Web-IO Digital Loggers can communicate with each other without involving an application program on a PC
- OPC
  W&T provides an OPC server which allows direct access to process and visualization applications to the Web-IO Digital Logger via OPC
- 50xxx
  Compatibility mode to older W&T Digital I/O Com-Servers

A detailed description for all the listed possibilities follows in the sections below.

5.4 Socket programming with command strings

Direct access via TCP or UDP sockets is possible for communicating with the Web-IO Digital Logger Digital Logger from simple application programs.

Data exchange between a PC and the Web-IO Digital Logger takes place here by passing simple command strings:

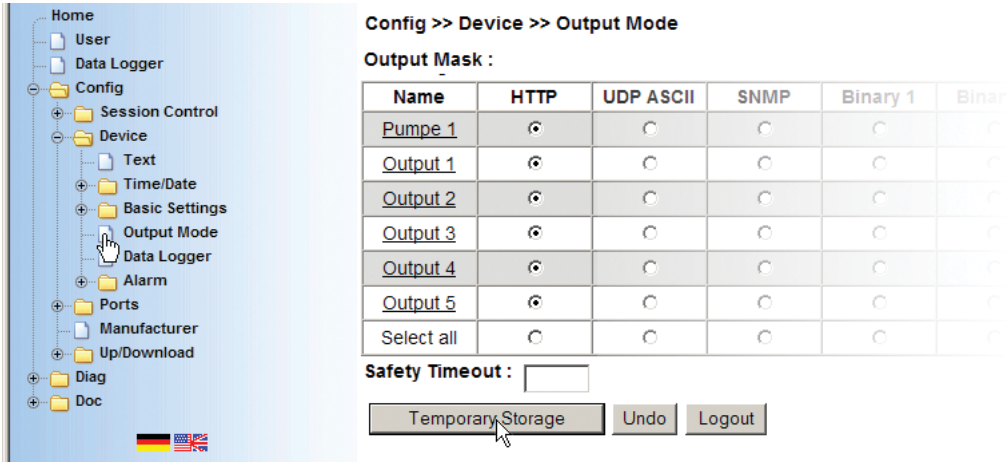
Commands	Parameter	Description
GET /input**x**	?PW=**password**& Password must be filled with the Administrator password. If no password was assigned, enter "PW=&". ***(applies to all commands!!)***	Input status request **x** can be a value between 0-11 and indicates the input. The reply from the Web-IO is a string beginning with "inputx;" followed by the input status: ON = Signal on input and OFF = no signal on input If x is completely omitted, the Web-IO returns a bit pattern corresponding to the input signals in hex format.
GET /counter**x**	?PW=**password**&	Counter value request **x** can be a value between 0-11 and indicates the input. The reply from the Web-IO is a string beginning with "counter**x**;". The counter state of the selected counter is attached in decimal format.
GET /outputaccess**x**	?PW=**password**& [Mask=XXXX&] State=**ON**/**OFF**/**YYYY**& [NA=ON&] ON: Output = 1, OFF: Output = 0, XXXX: Hex value between 0000 and 0FFF corresponding to the bits to be set YYYY: Hex-Wert between 0000 und 0FFF corresponding to the output bit map.	Setting one or more outputs **x** can be a value between 0-11 and indicates the output to be set. The reply from the Web-IO is a string beginning with "output;" followed by a bit map corresponding to the output signals, in hexadecimal format. Specifying the mask is optional. If mask is not sent, the outputaccess command applies to all outputs. "NA=ON" is an option for causing the Web-IO to send no reply to the outputaccess command.
GET /outputaccess**x**	?PW=**password**& State=**ON**/**OFF**/**XXXX**& ON: Output = 1, OFF: Output = 0, XXXX: Hex value between 0000 and 0FFF corresponding to the output bit pattern.	Sets one or more outputs **x** can be a value between 0-11 and indicates which output to set. The reply from the Web-IO is a string beginning with "output;" followed by a bit pattern corresponding to the output signals in hex format.
GET /counterclear**x**	?PW=**password**&	Resets a counter to 0. **x** can be a value between 0-11 and indicates which output to set. The reply from the Web-IO is a string beginning with "counter**x**;". The new counter state of the selected counter is attached in decimal format. If **x** is not specified, all 12 counters are reset to 0. In this case there is no reply.
GET /errorclear	?PW=**password**&	Clears the error report in the Web-IO. See section *Diagnostics and Testing*

Commands	Parameter	Description
GET /time	?PW=**password**&	Returns the system time of the Web-IO in format: DD.MM.YYYY,hh:mm:ss. D=day, M=month, Y=year, h=hour, m=minute, s=second
GET /settime	?PW=**password**& time=**DD.MM.YYYY,hh:mm:ss**&	Sets the system time of the Web-IO to the value transmitted with **time**
GET /diagnosis	?PW=**password**&	Requests the status of the diagnostic memory. The Web-IO returns: diagnosis;0000;00000000;00000000 The four-digit value represents the number of stored messages. For the 8-digit hexadecimal values each set bit represents one of the 64 possible messages.
GET /diagnosis**x**	?PW=**password**&	**x** indicates the index for one of the currently stored messages. The Web-IO returns the corresponding message text. **x** must not be larger than the number of currently saved messages.
GET /diaglist**x**	?PW=**password**&	Returns the messages for the individual message bits (max. 64)
GET /diagclear	?PW=**password**&	Clears the message memory
GET /errorclear	?PW=**password**&	Clear load errors and re-enables the affected outputs See section *Diagnostics and Testing*

5.4.1 TCP communication

First you must determine which of the 12 outputs will be made available for access via command strings.

Select in the navigation tree *Config >> Device >> Output Mode*



Required access rights: *Administrator*

All the outputs that will be used must be activated for *HTTP* (factory default).

If you have changed the output modes, click on the *Temporary Storage* button and then on the *Save and activate all temporary settings* link. Clicking on the *Save* button saves all your settings.

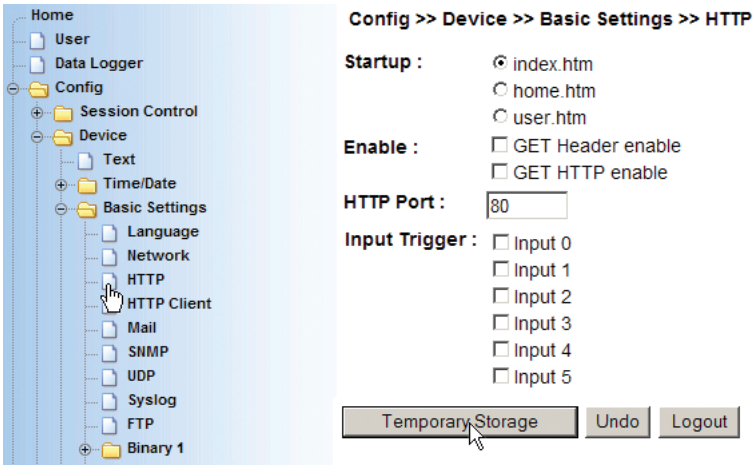
The Web-IO Digital Logger now functions as a TCP server, which results in a three-part program sequence regardless of which programming language is used:

1. Opening of the TCP connection
The application always acts as a client and therfore opens the TCP connection.

The socket interface of the Web-IO Digital Logger is broadly derived from HTTP protocol. This means that the **Listen port** is factory set to **TCP-Port 80** and corresponds to the configured HTTP port.

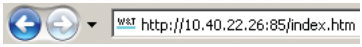
To use a different port, it is possible to change the HTTP port using Config >> Device >> Basic Settings >> HTTP.

m Required access rights: *Administrator*



Under HTTP port you may enter a TCP port for HTTP access which is different from the standard Port 80.

If the HTTP port is changed, the Web-IO Digital Logger will only be reachable from the browser by specifying the selected HTTP port.



Example for HTTP Port = 85.

This applies to all configuration accesses as well.

As a protection against blocked TCP connections, the Web-IO Digital Logger has a timer set for Port 80 which automatically closes the connection 30 seconds after it is opened if no syntactically correct commands have been received. After a connection has been successfully opened, the application should therefore always send for example an output query to the Web-IO Digital Logger.

2. Communication between Logger and application

Normal communication takes place in polling mode. This means: the client application uses the command strings to request the desired values and set the outputs.

Example 1: Query the status of Input 3, where there is a +12 V signal present. No administrator password was assigned for the Web-IO Digital Logger.

The application sends the following string to the Web-IO Digital Logger:

```
GET/input3?PW=&
```

The Web-IO Digital Logger responds as follows:

```
input3;ON
```

Example 2: Set outputs 1,3,4. In binary this is 0000 0000 0001 1010, which corresponds to 001A hex.

The administrator password for the Web-IO Digital Logger is: „blue“

The application sends the following string to the Web-IO Digital Logger:

```
GET/outputaccess?PW=blue&State=001a&
```

The Web-IO Digital Logger responds as follows:

```
output;001A
```

The Web-IO Digital Logger works with null-terminated strings! This means that everything the Web-IO Digital Logger sends to the application ends with a chr(0) *character*

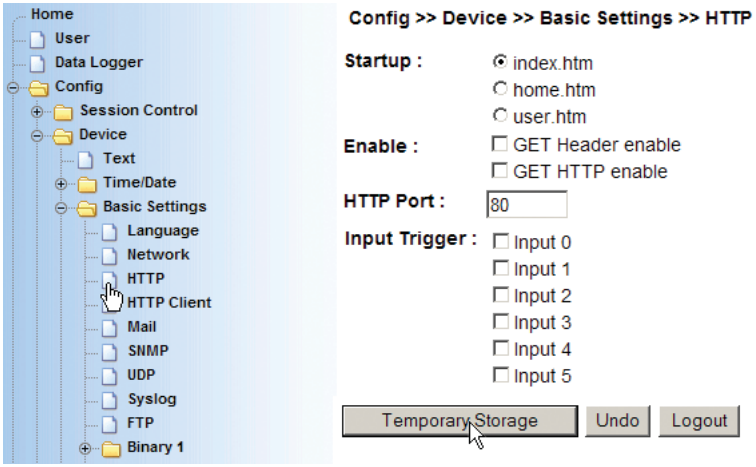
Event-driven messaging

Some applications make it necessary to report changes on certain inputs automatically over the existing TCP connection.

One reason for this is to reduce the data load on the network. If the inputs are polled at a cyclical interval, this will create a continuous data load. There is also the risk that changes on the inputs will not be detected if they occur between intervals.

To set the conditions for event-driven messaging, go to the navigation tree and select *Config >> Device >> Basic Settings >> HTTP*

Required access rights: *Administrator*



Input Trigger
Here you specify which inputs are to be monitored for a state change. If there is an existing connection, the Web-IO Digital Logger automatically sends the new input state to the application.

Include a header
The Web-IO Digital Logger can if needed precede the sent string with a header which contains the IP address and the name.

Example:

- The IP address of the Web-IO Digital Logger is 172.16.232.17.

- „W&T Web-IO Digital Logger 12xDigital“ was entered as the device name under *Config >> Device >> Text* .
- The application polled the inuts using *GET /input?PW=&*.

The Web-IO Digital Logger returns:

```
172.16.232.6;W&TWeb-IO62xDigital-logger;input;0000
```

If GET Header enable was activated, the header precedes all data transmission from the Web-IO Digital Logger.

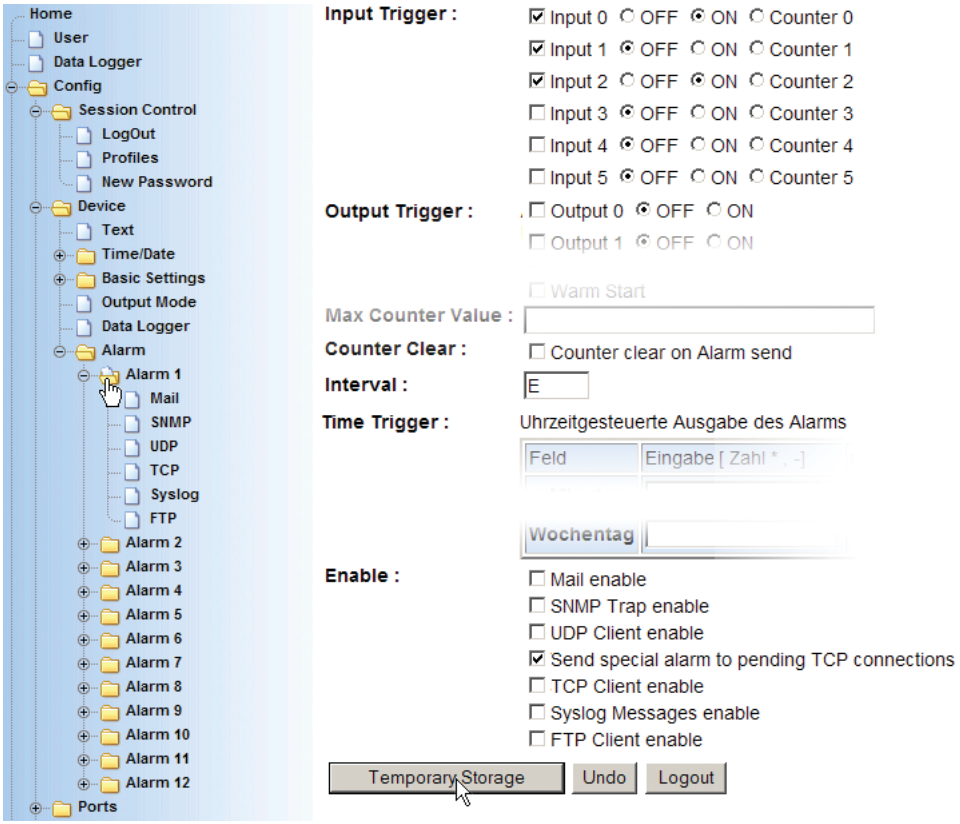
If all conditions have been specified, activate the new setting using *Config >> Session Control >> LogOut.*

Messaging with defined input patterns
In some applications it is practical to get the state of the inputs only when certain input patterns occur.

The messaging method described here can be used in tandem with normal event-triggered operation; in practice however this does not offer any advantages, since in this case there will be a message whenever there is a change. This means you would receive double messaging whenever the specified input pattern occurs.

To set the conditions for a TCP alarm message when certain input patterns occur, go to the navigation tree and select *Config >> Device >> Alarm >> Alarm 1*

Required access rights: *Administrator*



Input Trigger

Here you specify which inputs are to be monitored for which state. If you are activating multiple inputs in an alarm for monitoring, there is an AND-operation on the conditions.

Example:

In Alarm 1 Input 1 was activated for ON, Input 2 for OFF.

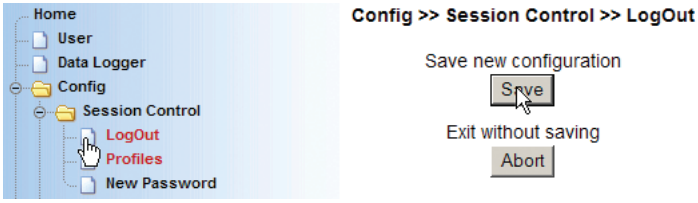
An alarm is only sent to the TCP application if both conditions are met. In this case: Input 1 = ON, Input 2 = OFF.

Interval
To send alarms over an existing TCP connection it is practical to set the *E* parameter here so that only one alarm message is sent when the alarm condition occurs.

Enable
Activate the *Send input to pending TCP connections* in the *Enable* area.

Pass the selected alarm conditions to the Web-IO Digital Logger by clicking on the *Temporary storage* button.

When all the alarm conditions have been specified, activate the new settings using *Config >> Session Control >> LogOut*



Clicking on the *Save* button restarts the Web-IO Digital Logger with the new parameters.

3. Close the TCP connection
Here the Web-IO Digital Logger deviates from HTTP protocol. In contrast to HTTP protocol, it is always the client application which closes a connection.

Exception: When it receives a defective command string, the Web-IO Digital Logger immediately closes the connection

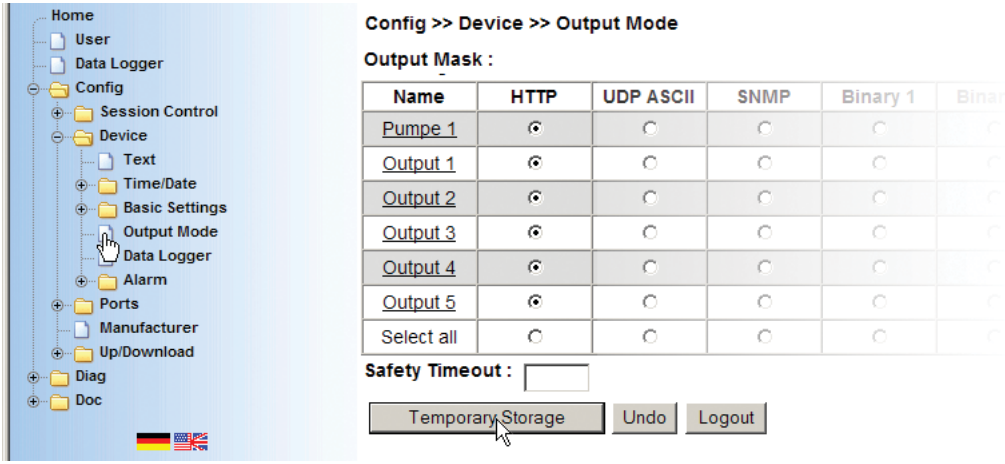
5.4.2 TCP communication - the Web-IO Logger as Client

In some applications it is necessary to use the Web-IO Digital Logger as client. In this case the Web-IO Digital Logger opens a TCP connection to a server application. The trigger for opening

the connection may be a status change on one of the inputs, or a time-based opening of a connection may be configured.

First you must specify which of the outputs should be available for access via command strings after the connection is opened.

In the navigation tree select *Config >> Device >> Output Mode*



Required access rights: *Administrator*

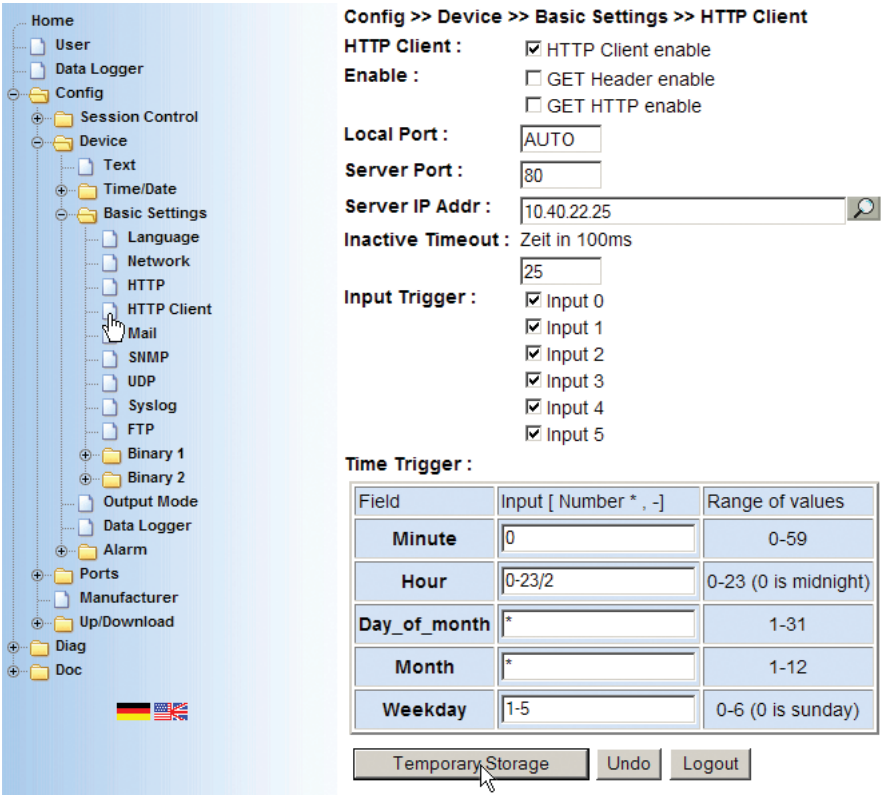
At least one or all outputs used must be activated for *HTTP*.

If you have changed the output modes, click on the *Temporary storage* button.

Then you must configure and activate the Web-IO Digital Logger for TCP-client communication.

In the navigation tree select *Config >> Basic Settings >> HTTP-Client*

Required access rights: *Administrator*



HTTP Client

This field must be acvitated for the Web-IO Digital Logger to operate as a TCP-client.

Enable

Activate this field if you want to precede the data sent by the Web-IO Digital Logger to the server application with a header.

Example:

- The IP address of the Web-IO Digital Logger is 172.16.232.17.
- As a *Device Name* „W&T Web-IO Digital Logger 12xDigital“ was entered under *Config >> Device >> Text*.
- The server application queried the inputs using *GET /input?PW=&*.

The Web-IO Digital Logger sends:

```
172.16.232.17;W&TWeb-IO12xDigital;input;0000
```

If *GET Header enable* was activated, the header precedes all the data sends from the Web-IO Digital Logger. In server applications that can accept more than one connection, this makes it very simple to distinguish from where the data is coming.

Local Port
Enter the desired local port in the *Local Port* field. The factory default setting is *AUTO* and the Web-IO Digital Logger uses the next available port.

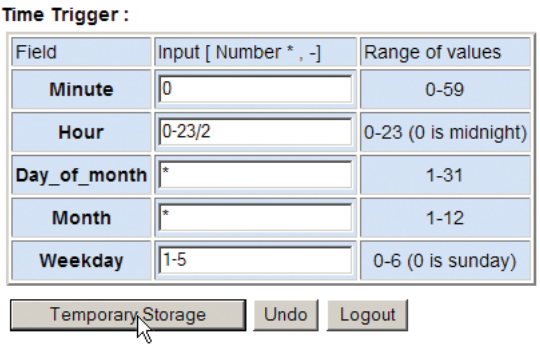
Server Port
Here you enter the port on which the server application accepts the connection.

Server IP-Adress
Here you enter the IP address of the PC to connect to.

Inactivity Timeout
After the timeout time entered here the Web-IO Digital Logger automatically closes the connection to the server. The timeout time starts as soon as there is no more data traffic on the connection.

Input Trigger
Here you highlight all inputs that are supposed to trigger a connection opening or input status packet for the server application when there is a status change.

Time Trigger
In this area you can configure a time-controlled connection opening.



The syntax for the time setting is derived from the Cron format used in Linux/Unix. Multiple entries separated by commas can be entered in one field.

The „*“ stands for any permissible whole-number value. Example: * in the Month field stands for every month.

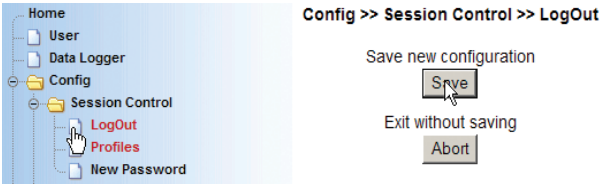
„-“ can be used to use time spans as triggers. Example: 1-5 in the Weekday field means Mo - Fr

If a „/“ is placed after the actual value or value range, a cycle value can be indicated behind it. Example: 0-23/2 in the Hour field means an alarm every two hours, even if the conditions in the other fields are met. Wildcards („*“) are not allowed together with „/“. (*/2 would not bring a reliable result)

The fields filled in as above therefore mean: every month, on every Monday through Friday, every two hours on the hour.

Send the selected settings to the Web-IO Digital Logger by clicking on the *Intermediate store* button.

Once all your settings have been made, activate them using *Config >> Session Control >> LogOut*



After clicking on the *Save* button the Web-IO Digital Logger will now run as a TCP client.

Communication between Web-IO Logger and application

1. Opening the TCP connection

The TCP connection is opened by the Web-IO Digital Logger according to the configured trigger conditions.

After the connection has been successfully opened, the Web-IO Digital Logger sends the input status in format

```
input;0000
```

to the server application.

The 4-place numerical value (8-place for the 24x Web-IO Digital Logger) represents the input status in hex format.

The Web-IO Digital Logger uses null terminated strings! This means that everything the Web-IO Digital Logger sends to the application ends with a 0 character.

For Web-IO Digital Loggers with relay outputs you must maintain a pause of at least 200ms between two switching operations on one output!

2. Communication between Logger and application

Whenever there is an existing connection, the server application can communicate with the Web-IO Digital Logger using command strings.

Example 1: Query the status of Input 3, on which a +12 signal is present. No administrator password has been assigned to the Web-IO Digital Logger.

The application sends the following string to the Web-IO Digital Logger:

```
GET/input3?PW=&
```

The Web-IO Digital Logger returns with:

```
input3;ON
```

Example 2: Set outputs 1,3,4 In binary this is 0000 0000 0001 1010, which corresponds to 001Ah.

The administrator password for the Web-IO Digital Logger is: „blue“

The application sends the following string to the Web-IO Digital Logger:

```
GET/outputaccess?PW=blau&State=001A&
```

The Web-IO Digital Logger returns with:

```
output;001A
```

The Web-IO Digital Logger uses null terminated strings! This means that everything the Web-IO Digital Logger sends to the application ends with a 0 character.

3. Closing the TCP connection

The Web-IO Digital Logger automatically closes the connection when the set timtout time after the last data communication between Web-IO Digital Logger and server application has elapsed.

Naturally the server application may close the connection at any desired time.

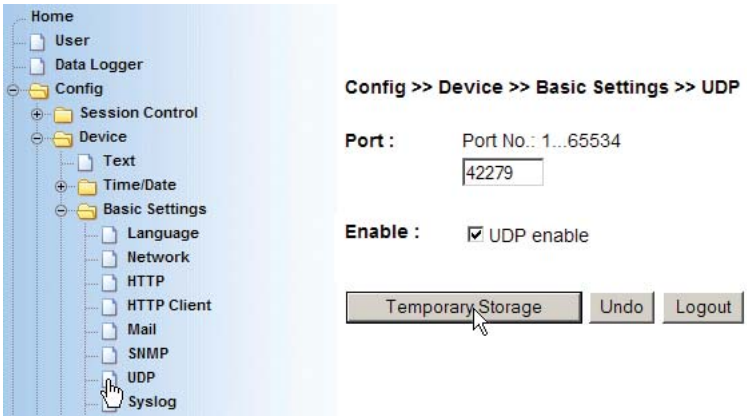
5.4.3 UDP communication

First you need to activate the Web-IO Digital Logger for communication via UDP command strings and enter the local port.

Select in the navigation tree *Config >> Basic Settings >> UDP*

Required access rights: *Administrator*

In the Port field enter the desired local port. The factory setting is Port 42279. Check UDP enable.



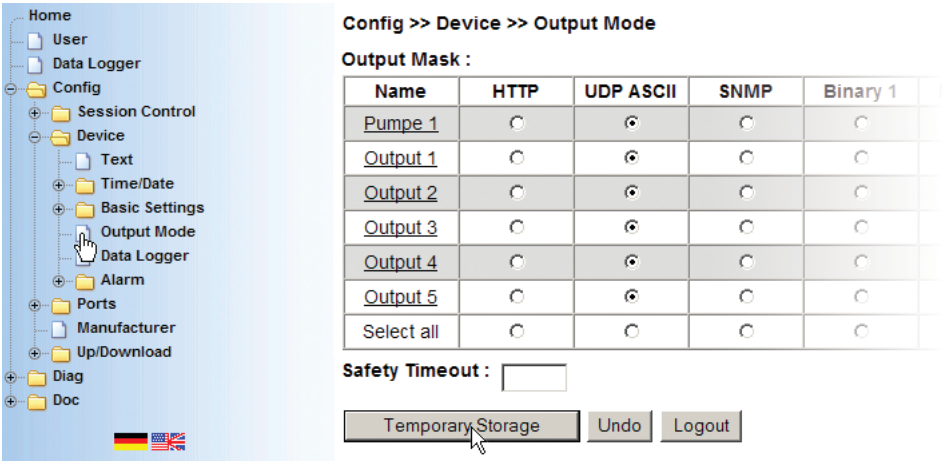
Click on the *Temporary Storage* button to pass the selected settings to the Web-IO Digital Logger.

You must also specify which of the 12 outputs you want to make available for access via UDP using command strings.

Select in the navigation tree *Config >> Device >> Output Mode.*

Required access rights: *Administrator*

All the outputs used must be activated for *UDP ASCII*.



If you have changed the output modes, click on the *Temporary Storage* button and then on the *Save and activate all temporary settings* link. Clicking on the *Save* button saves all your settings.

Communication between Logger and the application

In contrast to TCP, there is no communication between the Web-IO Digital Logger and the other communications partner when communicating via UDP protocol. Data are exchanged using datagrams.

Normal communication takes place in polling mode. This means the application uses command strings to request the desired values and set the outputs.

The Web-IO Digital Logger always responds to a poll by returning the IP address and port of the requesting network station.

Example 1: Query the status of Input 3, where there is a signal of +12 V present. No administrator password was asigned for the Web-IO Digital Logger

The application sends the following string to the Web-IO Digital Logger:

```
GET/input3?PW=&
```

The Web-IO Digital Logger responds as follows:

```
input3;ON
```

Example 2: Set outputs 1,3,4 in binary this is 0000 0000 0001 1010 , which corresponds to 001A hex.

The administrator password for the Web-IO Digital Logger is: „blue“

The application sends the following string to the Web-IO Digital Logger:

```
GET/outputaccess?PW=blue&State=001A&
```

The Web-IO Digital Logger responds as follows:

```
output;001A
```

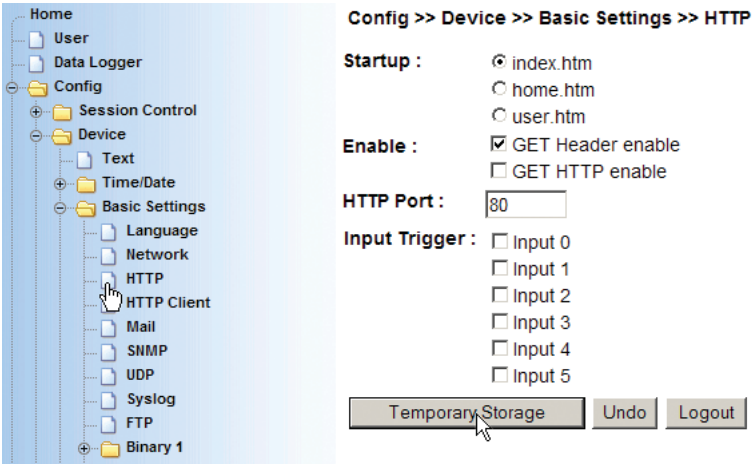
The Web-IO Digital Logger works with null-terminated strings! This means that everything the Web-IO Digital Logger sends to the application ends with a chr(0) *character*

So that under UDP a message can be automatically received when states change, the Web-IO Digital Logger provides a variety of various alarm conditions.

A detailed description of how to use these alarm possibilities can be found in the Alarms *section*

Include a header

The Web-IO Digital Logger can if needed precede the sent string with a header containing the IP address and the name.



To enable a header, go to *Config >> Basic Settings >> HTTP* and select *GET Header enable*. Clicking on *Logout* and *Save* activates sending of the header.

Example:

- The IP address of the Web-IO Digital Logger is 172.16.232.17.
- „W&T Web-IO 12xDigital“ was entered as the device name under *Config >> Device >> Text* „“.
- *GET /input?PW=&* was used by the application to poll the inputs.

The Web-IO Digital Logger returns:

172.16.232.6;W&T Web-IO 12xDigital;input;0000

If GET Header enable was activated, the header precedes all data sent by the Web-IO Digital Logger.

5.4.4 Program-controlled error management

If the Web-IO Digital Logger detects one or more errors while processing its tasks, these are saved and are viewable from the navigation tree under *Diag*.

A more detailed description of this can be found in Section 6.6 Diagnostics and Testing.

In some applications it may be desired that error management is handled not manually by viewing a Web page, but rather errors are automatically evaluated in a program.

For such cases the Web-IO Digital Logger provides several command strings for using TCP or UDP.

Creating a list pf possible error messages

The Web-IO Digital Logger can manage a maximum of 64 different error types. The actual number may vary within this limit depending on the firmware version.

To get an overview of the possible errors, the command

```
GET/diaglistx?PW=password&
```

can be sent to the Web-IO Digital Logger.

x is replaced by a value between 0 and 64. INstead of *password* enter the Administrator or Operator password.

The Web-IO Digital Logger replies with the corresponding error message.

```
diaglistx;errortext
```

Example:

If the application sends (Passwort = „wut“) to the Web-IO Digital Logger

```
GET/diaglist2?PW=wut&
```

the Web-IO Digital Logger responds with

```
diaglist2;FormatfehlerinderDNSAnfrage
```

With a *for next* loop the *diaglist* command can be used to read out all possible error messages and save them as a static list. In binary mode, for example, you can only determine how many errors there are currently. The associated error numbers are also sent. By using the static error list a readable error can be output.

Direct evaluation of the current errors

To get the current error status of the Web-IO Digital Logger you can use the command

```
GET/diagnosis?PW=password&
```

The Web-IO Digital Logger responds with

```
diagnosis;zzzz;yyyyyyyy;xxxxxxxx
```

In *zzzz* the Web-IO Digital Logger returns the number of current errors in 4-place hex format.

yyyyyyyy; xxxxxxxx may be ignored for normal, standard applications.

For expert programmers: yyyyyyyy; xxxxxxxx *are two 32-bit values, each in 8-place hex format. THe resulting 64 bits substitute for the 64 possible error types. By using the static error list (see GET /diaglist...) the individual bits can be decoded. The LSB appears to the right in the x range and the MSB to the left in the y range.*

To get the current errors as an error text, the Web-IO Digital Logger provides the command

```
GET/diagnosisx?PW=&
```

Here x is the index for the error in decimal format beginning with 1.

The reply from the Web-IO Digital Logger then looks as follows:

```
diagnosisx;errortext
```

In contrast to the error texts which are read from the Web-IO Digital Logger using *diaglist*, the error text for *diagnosis* may also contain dynamic elements such as IP addresses or port numbers.

Example:

The application sends:

```
GET/diagnosis?PW=wut&
```

The Web-IO Digital Logger responds:

```
diagnosis;0005;00400040;008A0000
```

The application sends:

```
GET/diagnosis1?PW=wut&
```

The Web-IO Digital Logger responds:

```
diagnosis1;MailServerantwortetnicht.
```

The application sends:

```
GET/diagnosis2?PW=wut&
```

The Web-IO Digital Logger responds:

```
diagnosis2;DasVersendenderMailwurdeabgebrochenundwirdwiederholt.
```

The application sends:

```
GET/diagnosis3?PW=wut&
```

The Web-IO Digital Logger responds:

```
diagnosis3;ZielIP-Adresseunbekannt:172.16.232.8.
```

The application sends:

```
GET/diagnosis4?PW=wut&
```

The Web-IO Digital Logger responds:

```
diagnosis4;WatchdogTimerabgelaufen!
```

The application sends:

```
GET/diagnosis5?PW=wut&
```

The Web-IO Digital Logger responds:

```
diagnosis5;TCPClientAlarm:Servernichterreicht.
```

Clearing the error memory

The list of errors occuring during runtime remains stored in the Web-IO Digital Logger, even if the error is no longer present at the time of query.

To clear the error memory, use the command

```
GET/diagclear?PW=password&
```

The Web-IO Digital Logger responds with

```
diagnosis;0000;00000000;00000000
```

if there are no current errors.

5.5 BINARY - socket programs with binary structures

The Web-IO 12 X Digital provides two independent socket accesses, *Binary 1* and *Binary 2*, for carrying out binary data exchange. Both can be used and configured independently of each other.

Whether the Web-IO Digital Logger functions on the respective BINARY socket as a TCP server, TCP client or UDP peer depends on the desired application.

First an overview of which operation mode the Web-IO Digital Logger uses for which application.

- Your own socket application (binary with password protection)
  - TCP-Server
  - TCP-Client
  - UDP-Peer
- Your own socket application, which was originally programmed for the W&T Digital I/O Com-Server 50xxx.
  - TCP-Server
  - TCP-Client
  - UDP-Peer
- Box-to-Box Master
  - TCP-Client
- Box-to-Box Slave

- TCP-Server

- OPC-Device together with the W&T OPC-Server
  - TCP-Server

Binary socket access and 50xxx-Mode
In this section you will be shown how the Web-IO 12XDigital can be accessed from your own professional applications using sockets with binary structures.

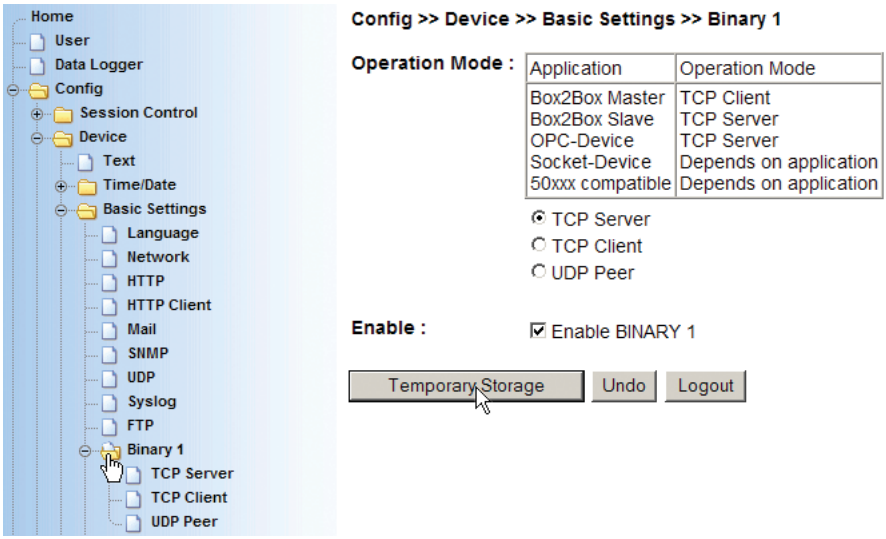
Box-to-box and OPC device modes will be treated in detail in the following sections

5.5.1 Specifying the operation mode

First you must specify whether the Web-IO Digital Logger is to be a TCP client, TCP server or UDP peer in your application.

Select in the navigation tree *Config >> Device >> Basic Settings >> Binary 1* if you want to configure the operation mode for access through *Binary 1*.

Required access rights: *Administrator*



After selecting the desired mode, pass the setting to the Web-IO Digital Logger by clicking on *Temporary Storage*.

Two levels of socket programming are available to the programmer for access from your own application programs.

1. Password protected access

2. 50xxx-Mode compatible with the older W&T Digital I/O Com-Servers

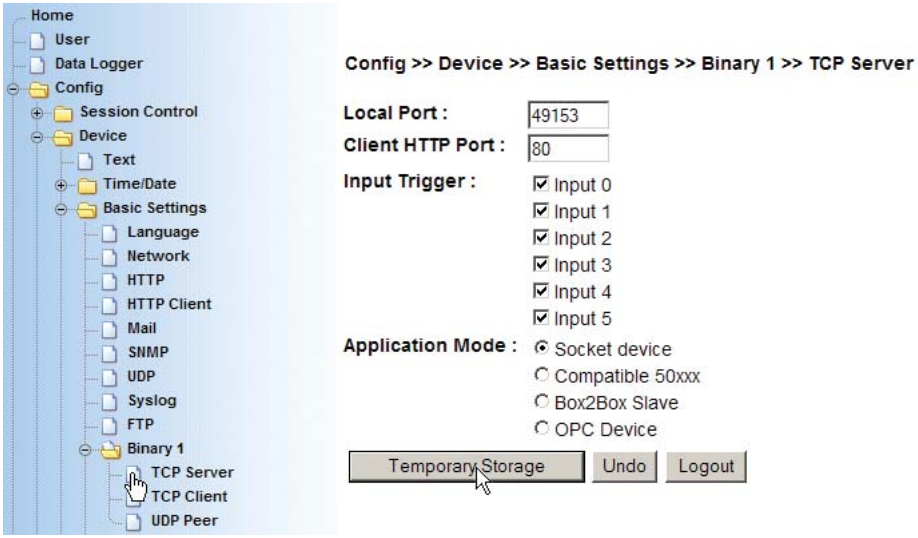
Both levels work with the same binary structures and differ only in the lack of password protection in 50xxx mode.

5.5.2 The Web-IO Digital Logger as socket server

To operate the Web-IO Digital Logger as a socket server, you must make a few additional settings.

Select in the navigation tree *Config >> Device >> Basic Settings >> Binary 1>> TCP Server*

Required access rights: *Administrator*



Local Port
The local port of the Web-IO Digital Logger is factory set for 49153. If your application requires a different local port for the Web-IO Digital Logger, enter the desired port number in the *Local Port* field.

Client HTTP Port
Has meaning only in OPC and Box2Box modes and specifies the HTTP port on which the control connection to the OPC server or slave box should be opened.

Unless otherwised specified, always use Port 80 here.

Input Trigger
Here you check the inputs whose state change will trigger data to be sent to the client application (important for event-triggered applications).

Application Mode
Select here:

- *Default* - If you want access to the Web-IO Digital Logger to be password protected.
- *Compatible 50xxx* - If you want to access applications that were programmed for the older Digital I/O Com-Servers. You can also use this mode for new applications that do not require password protection.

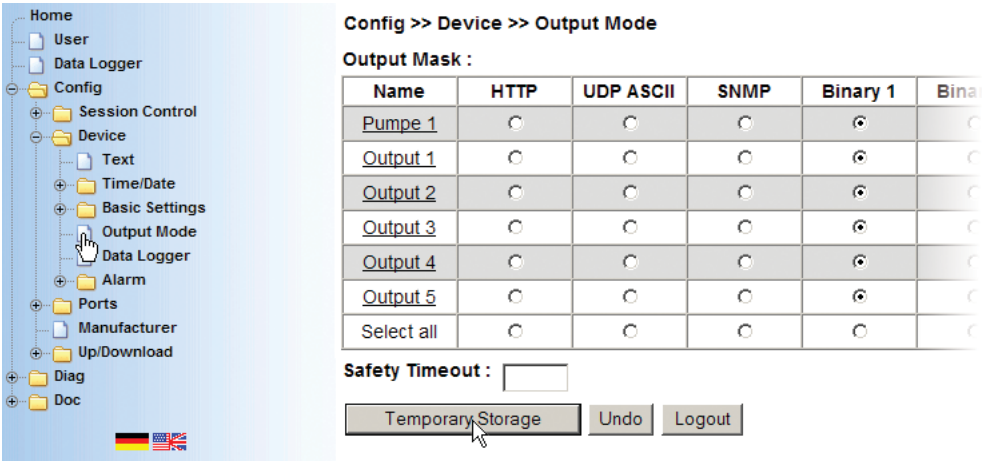
Additional information concerning *Box2Box Slave* and *OPC Device* modes can be found in the corresponding sections.

Once you have made all your settings, send them to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

In addition you must enable the corresponding outputs for binary mode.

Under *Config >> Device >> Output Mode* select the desired bnary access.

Required access rights: *Administrator*



Once you have made all your settings, send them to the Web-IO Digital Logger by clicking on the *Logout* button. Clicking on the *Save* button activates the settings.

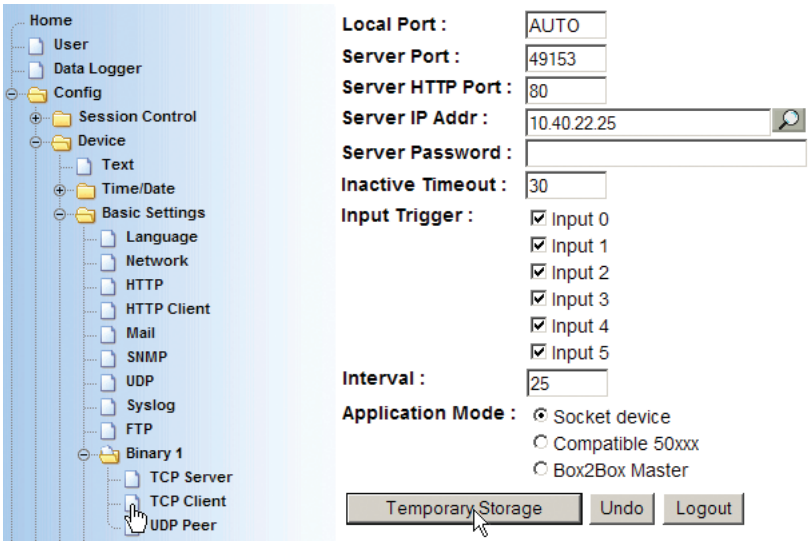
All the configuration possibilities shown for Binary 1 *can also be used for* Binary 2.

5.5.3 The Web-IO Digital Logger as socket client

To operate the Web-IO Digital Logger as a socket client, you must make a few additional settings.

Select in the navigation tree *Config >> Device >> Basic Settings >> Binary 1 >> TCP Client*

Required access rights: *Administrator*



Local Port
The local port of the Web-IO Digital Logger is factory set for 49153. If your application requires a different local port for the Web-IO Digital Logger, enter the desired port number in the *Local Port* field.

Server Port
Enter here the port number which the server application will use to accept the connection.

Server HTTP Port
Has meaning only in OPC and Box2Box modes and specifies the HTTP port on which the control connection to the OPC server or slave box should be opened.

Unless otherwised specified, always use Port 80 here.

Server IP Addr
Enter here the IP address of the server.

Server Password
A server password only needs to be entered if you want the Web-IO Digital Logger to work as a box-to-box master or access

an additional Web-IO Digital Logger as a TCP client in server mode. See Box-to-Box section.

Inactive Timeout
Here you configure the timer used by the Web-IO Digital Logger to close the connection after the time has expired. Enter the value in decimal using 100ms increments. When there is an active network connection the timer is reset once all the data have been exchanged.

Example: The value 10 corresponds to one second. If no data transfer is detected during this time, the Web-IO Digital Logger closes the connection.

Entering no value means automatic connection closing is deactivated.

Input Trigger
Check the inputs whose state change you want to use as triggers for opening the TCP connection and sending data to the server (important for event-triggered applications).

Interval
If you want to cyclically notify the server application of input state changes, enter the interval here in (value will be multiplied with 100ms) .

Example: The value 300 corresponds to 30 seconds.

Please note that when using charged dial-up connections, too short an interval can result in this connection remaining open while charges continue to accumulate!

Mode
Select here:

- *Default* - If you want access to the Web-IO Digital Logger to be password protected.
- *Compatible 50xxx* - If you want to access applications that were programmed for the older Digital I/O Com-Servers. You

can also use this mode for new applications that do not require password protection.

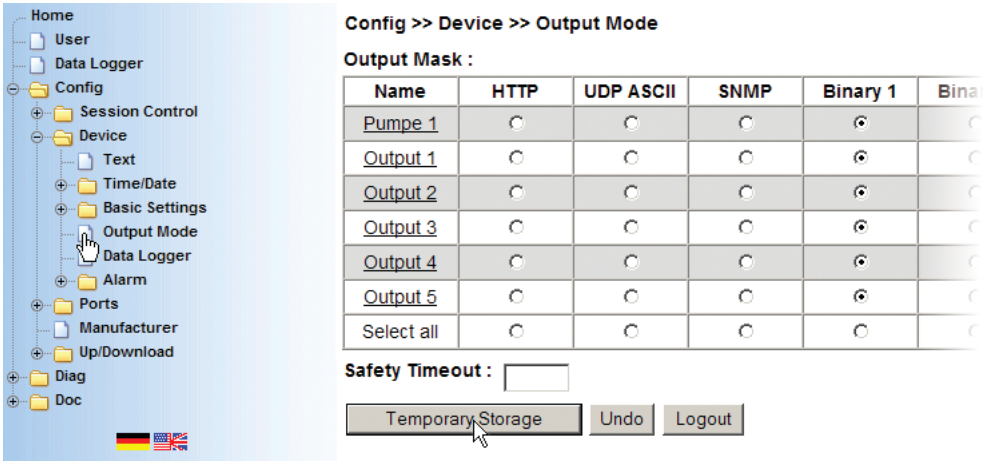
Additional information about *Box2Box Master* can be found in the section *Box-to-Box.*

Once you have made all your settings, send them to the Web-IO Digital Logger by clicking on *Temporary Storage*.

In addition you must enable the outputs you want to use for binary mode.

Under *Config >> Device >> Output Mode* check the desired binary access.

Required access rights: *Administrator*



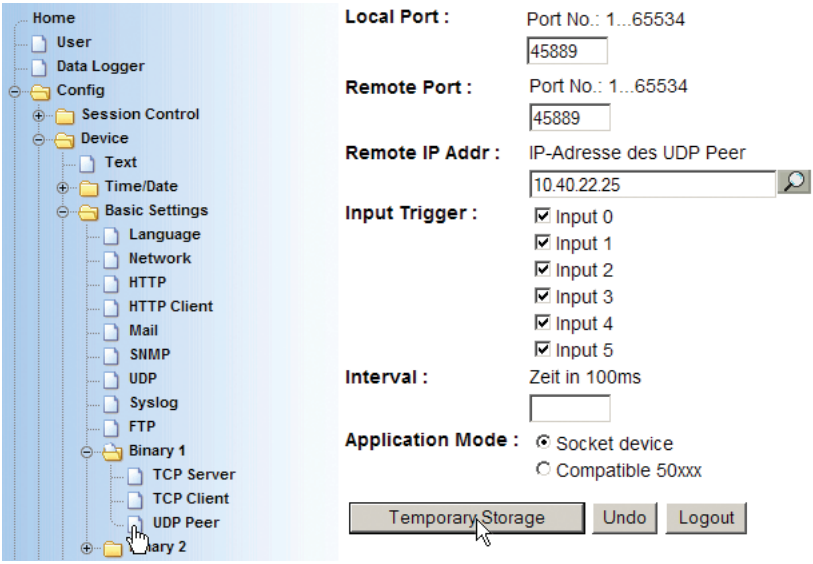
Once you have made all your entries, send them to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.. Clicking on the *Save* button activates the settings.

5.5.4 The Web-IO Digital Logger as UDP Peer

To operate the Web-IO Digital Logger as a UDP peer, you must make a few additional settings.

Select in the navigation tree *Config >> Device >> Basic Settings >> Binary 1>> UDP Peer*

Required access rights: *Administrator*



Local Port
The local port of the Web-IO Digital Logger is factory set for 45889. If your application requires a different local port for the Web-IO Digital Logger, enter the desired port number in the *Local Port* field.

Remote Port
Enter here the port number the UDP application communicating with the Web-IO Digital Logger will use to receive data.

Remote IP Addr
Enter here the IP address of the communications partner.

Input Trigger
Check here the inputs whose state change will be used as a trigger for sending a UDP datagram (important for event-triggered applications).

Interval
If you want to send the state of the inputs cyclically to the communications partner, the interval can be entered here in 100ms increments.

Example: The value 300 corresponds to 30 seconds.

Please note that when using charged dial-up connections, too short an interval can result in this connection remaining open while charges continue to accumulate!

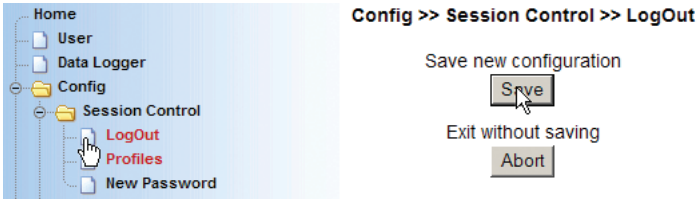
Application Mode
In UDP peer configuration, there is no difference between the Default and Compatible 50xxx modes.

Once you have made all your settings, send them to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

In addition you must enable the corresponding inputs for binary mode.

Under *Config >> Device >> Output Mode* select the desired binary access.

Required access rights: *Administrator*



Once you have made all your entries, send the setting to the Web-IO Digital Logger by cliking on the *Logout* button. Clicking on the *Save* button activates the settings.

5.5.5 Passwort protection

As already mentioned several times, the Web-IO Digital Logger in TCP server mmode makes it possible to protect access to the application by means of a password..

Before the actual connection to the Web-IO Digital Logger is opened, the *BIN_INFO* structure defined here must be sent to Port 80 of the Web-IO Digital Logger over a separate TCP connection.

The Web-IO Digital Logger also uses the *BIN_INFO* structure for its reply.

BinInfo (PC <-> Web-IO)			
	BYTE[n]	HTTPlogin	n = 14 Byte + Password
	WORD	dummy	always 0
	BYTE	type	type of request
	BYTE	subtype	additional type information
	LONG	srcip	source IP address
	WORD	srcport	source port
	WORD	destport	detination port

The individual variables of the structure are filled as follows::

HTTPLogin[n]
Is a byte field or string consisting of a login string and the administrator password..

```
GET/bin?LPW=<AdministatorPasswort>&
```

n stands for the number of bytes used and corresponds to 14 + the length of the password. The length of the password is restricted to max. 31 characters.

In the reply from the Web-IO Digital Logger HTTPLogin is always 8 characters long and contains the following string::

```
GET/bin
```

Dummy
Separator between the ASCII and binary part of the structure..

is always = 0x00

Type
Determines the type in which the binary mode shall be used.

The application must enter 0x04 here to open a TCP connection.

In its reply the Web-IO Digital Logger enters

0x02 if the connection request was accepted.

0x03 if the connection request was rejected..

SubType
Gives additional details on the status of the connection request..

The application always sends 0x00.

The Web-IO Digital Logger replies with

```
0x01    BINSUBTYPE_OK,          // if the connection request was accepted..
0x02    BINSUBTYPE_NO_ACCESS,   // if a connection is already open
0x04    BINSUBTYPE_WAIT,        // If the connection is only allowed to be opened
                                // after a timeout
0x07    BINSUBTYPE_PW_MISMATCH,           // if wrong password
0x08    BINSUBTYPE_DEST_PORT_MISMATCH,    // if wrong Destination Port
0x09    BINSUBTYPE_MODUS_MISMATCH,        //ifwrongmode
```

If 0x01 or 0x04 was received, the actual data connection can be opened.

SrcPort
The client application always enters a 0 here.

The Web-IO Digital Logger returns here the opened server port (e.g., 49153 for binary 1). If the login attempt fails, the Web-IO Digital Logger enters 80.

DestPort
The client application enters here which port the connection is to be opened on. (e.g. 49153 for Binary 1 or 49154 for Binary 2)

The Web-IO Digital Logger always returns 0.

The connection through which the BinInfo structure was exchanged is automatically closed by the Web-IO Digital Logger.

5.5.6 BINARY - The IO structures

To enable simply communication between the application program on the computer and the Web-IO Digital Logger, there is a variety of structures (variable fields) which define the format and content of the data which are exchanged between the application program and the Web-IO Digital Logger.

IO structures are offered for the following functions:

- Reading inputs
- Reading inputs and outputs
- Setting outputs
- Reading counters
- Resetting counters
- Parameterizing cyclical and automatic messaging for state changes.

The application program employs the easy-to-use socket interface (Windows: WinSock, UNIX, Linux: Berkley Sockets) for exchanging data with the Web-IO Digital Logger over the network via TCP/IP in the form of these data structure.

The IO structures are independent of the selected network protocol (TCP or UDP).

The choice of which protocol to use, UDP or TCP, depends on the type of application. Both protocols have their advantages and disadvantages that need to be weighed depending on the application you are creating.

Help for socket programming including the basics of TCP/IP can be found in a short, easy to understand guide „Ready for TCP/IP Sockets in 1 Day". Program examples for client/server applications under TCP/IP can be downloaded from our homepage at http://www.wut.de.

5.5.7 Definition of the IO structures

In order to uniquely identify and process the contents of a packet in BINARY mode, all the data must be sent to the Web-IO Digital Logger in the form of these IO structures, regardless of whether you are using *50xxx-compatible* or *Default* mode.

All structures begin with the same header, which consists of the following 4 WORDs (16bit Integer):

send_sequence and *rec_sequence* are only filled if you want to implement control of the packet sequence yourself in UDP mode. Otherwise these values are always 0 and must not be omitted!

The value *struct_type* identifies the contents of the structure after the first 4 WORDs and indicates the total length of the structure in bytes, i.e., including the first 4 WORDs. This results in the following general packet structure:

Struktur Construction			
	WORD	send_sequenz	always 0
	WORD	rec_sequenz	always 0
	WORD	struct_type	identification of structur
	WORD	length	length of structure
	VARIBLE	.....	further variables
	.....	.....	further variables

Note: The following applies to all IO structures:

A **word** corresponds to 16bit integer.

A **char** corresponds to one byte (8bits)
A **long** corresponds to a 32bit integer

Hexadecimal format **0x** in front of the value

For sending and receiving the variable **Low-Byte first** applies to all structure variables.

The following structure

Beispiel

WORD	send_sequenz	0x0000
WORD	rec_sequenz	0x0000
WORD	struct_type	0x0001
WORD	length	0x0008

would look as follows when sending over the network.

send_sequenz		rec_sequenz		struct_type		length	
low byte	high byte	low byte	high byte	low byte	high byte	low byte	high byte
00	00	00	00	01	00	08	00

5.6.8 Working with the IO structures

In the next sections the individual structures used for *name* are declared and explained and the corresponding values of the variables *send_sequence*, *rec_sequence*, *struct_type* and *length*, with which each packet begins, are given.

The IO structure ReadRegister
Sending this structure to the Web-IO Digital Logger causes it to send the status of Inputs 0 - 11 to the application program. The packet consists only of these four WORDs. This structure is used by the user program and the Web-IO Digital Logger always responds by sending the WriteRegister structure.

ReadRegister
(PC -> Web-IO)

WORD	send_sequenz	immer 0
WORD	rec_sequenz	immer 0
WORD	struct_type	0x0001
WORD	length	0x0008

The IO structure WriteRegister
This structure is used to send the state of the inputs or outputs for the Web-IO 12xDigital. If the application program sends this

structure to the Web-IO Digital Logger, the Web-IO Digital Logger sets the outputs corresponding to the value transmitted in *value*.

If the Web-IO Digital Logger sends this structure to the user program, *value* has the value corresponding to the input state.

WriteRegister (PC <-> Web-IO)			
	WORD	send_sequenz	immer 0
	WORD	rec_sequenz	immer 0
	WORD	struct_type	0x0008
	WORD	length	0x000C
	WORD	word_anz	immer 1
	WORD	value	Input Status oder Output Status

For Web-IO Digital Loggers with relay outputs you must maintain a pause of at least 200ms between two switching operations on one output!

The IO structure SetBit

This structure allows you to set individual outputs on the Web-IO 12xDigital. If for example the entire process status is not represented in the user program, individual output can be set without changing the value of the others. Bits 0..11 in *set_bits* and *value* correspond to the respective outputs. This structure is used only by the application program.

SetBit (PC -> Web-IO)			
	WORD	send_sequenz	immer 0
	WORD	rec_sequenz	immer 0
	WORD	struct_type	0x0009
	WORD	length	0x000C
	WORD	set_bits	Maske der zu setzenden Bits
	WORD	value	Wert der zu setzenden Bits

Example.:

```
set_bits=0x0124/value=0x0104
```

Output 2 and Output 8 (counting Output0..11) are set to ON and Output 5 to OFF. No other outputs are changed.

The IO structure RegisterRequest

This structure sends the user program to the Web-IO 12xDigital so that the contents of inputs and outputs can be read in

overview form. The Web-IO 12xDigital always responds with the I/O structure *RegisterState*

RegisterRequest (PC -> Web-IO)			
	WORD	send_sequenz	immer 0
	WORD	rec_sequenz	immer 0
	WORD	struct_type	0x0021
	WORD	length	0x0008

The IO structure RegisterState

The Web-IO 12xDigital sends the contents of the inputs and outputs with this structure. This structure is only sent if the user program has sent the structure *Register Request* to the Web-IO Digital Logger.

RegisterState (PC <- Web-IO)			
	WORD	send_sequenz	immer 0
	WORD	rec_sequenz	immer 0
	WORD	struct_type	0x0031
	WORD	length	0x000E
	WORD	driver_id	immer 2
	WORD	input_reg	Input Status
	WORD	output_reg	Output Status

The IO structure Send Mode

This structure is used to specify the trigger conditions the Web-IO 12xDigital uses to send the status of the inputs to the user program. There are basically three possibilities, but they may be combined with each other:

1. The user program polls the Web-IO Digital Logger by sending the READ structure
2. The Web-IO Digital Logger sends the WriteRegister - structure with the status of the inputs in a configurable interval
3. The Web-IO Digital Logger sends the WriteRegister - structure with the status of the inputs after a state change of the configured inputs

SendMode (PC -> Web-IO)			
	WORD	send_sequenz	immer 0
	WORD	rec_sequenz	immer 0
	WORD	struct_type	0x0010
	WORD	length	0x000C
	WORD	ea_bits	Maske für Inputtrigger
	WORD	interval	Intervall für Datenpakete in 100ms

The IO structure ReadCounter

The user program sends this structure to the Web-IO Digital Logger to request the counter state of a certain input counter. Which input is intended is sent in the variable *counter_index*. The Web-IO Digital Logger always replies with the *COUNTER* structure.

ReadCounter (PC -> Web-IO)			
	WORD	send_sequenz	always 0
	WORD	rec_sequenz	always 0
	WORD	struct_type	0x00B0
	WORD	length	0x000A
	WORD	counter_index	Input 0..11 (23 for Web-IO 24)

The IO-Structure ReadClearCounter

The application program sends this structure to the Web-IO Digital Logger in order to request the counter status of a particular input counter and then immediately set to counter to 0. Which input this involves is transmitted in the variable *counter_index*. The Web-IO Digital Logger always replies with the structure *COUNTER*.

ReadClearCounter (PC -> Web-IO)			
	WORD	send_sequenz	always 0
	WORD	rec_sequenz	always 0
	WORD	struct_type	0x00C0
	WORD	length	0x000A
	WORD	counter_index	Input 0..11 (23 for Web-IO 24)

The IO structure Counter

With this structure the Web-IO Digital Logger sends the counter state of the input counter specified in *counter_index*.

Counter (PC <- Web-IO)			
	WORD	send_sequenz	always 0
	WORD	rec_sequenz	always 0
	WORD	struct_type	0x00B4
	WORD	length	0x000E
	WORD	counter_index	Input 0..11 (23 for Web-IO 24)
	LONG	counter_value	counter value

The IO structure ReadAllCounter

The user program sends this structure to the Web-IO Digital Logger to request the counter states of all the inputs in a data packet. The Web-IO Digital Logger always replies with the structure *ALL_COUNTER*.

ReadAllCounter (PC -> Web-IO)	WORD	send_sequenz	always 0
	WORD	rec_sequenz	always 0
	WORD	struct_type	0x00B1
	WORD	length	0x0008

The IO-Structure ReadClearAllCounter

The application program sends this structure to the Web-IO Digital Logger in order to request the counter states of all inputs in a data packet and then immediately sets the counters to 0. The Web-IO Digital Logger always replies with the structure *ALL_COUNTER*.

ReadClearAllCounter (PC -> Web-IO)	WORD	send_sequenz	always 0
	WORD	rec_sequenz	always 0
	WORD	struct_type	0x00C1
	WORD	length	0x0008

The IO structure AllCounter

The Web-IO Digital Logger uses this structure to send the counter states of all the inputs at one time.

AllCounter (PC <- Web-IO)	WORD	send_sequenz	always 0
	WORD	rec_sequenz	always 0
	WORD	struct_type	0x00B5
	WORD	length	0x003A for Web-IO 12xDigital 0x006A for Web-IO 24xDigital
	WORD	counter_count	number of Counters
	LONG	counter_value	Counter 0
	LONG	.....	..
	LONG	counter_value	Counter 11 (23 for Web-IO 24)

The IO structure ClearCounter

The user program sends this structure to the Web-IO Digital Logger to reset the counter state of a certain input counter to 0. Which input is intended is sent in the variable *counter_index*

ClearCounter (PC -> Web-IO)	WORD	send_sequenz	always 0
	WORD	rec_sequenz	always 0
	WORD	struct_type	0x00B2
	WORD	length	0x000A
	WORD	counter_index	Input 0..11 (23 for Web-IO 24)

IO-Structure ReadDiagnosis

If the Web-IO Digital Logger determines that there is a communications or system error, the latter is listed on the HTML page *diag* and can be read using the browser. Since error

management via browser for program-controlled applications is not always available, the error status of the Web-IO Digital Logger can be queried using the structure *ReadDiagnosis*

ReadDiagnosis (PC -> Web-IO)			
	WORD	send_sequenz	always 0
	WORD	rec_sequenz	always 0
	WORD	struct_type	0x00D1
	WORD	length	0x0008

The Web-IO Digital Logger replies with a Diagnosis type structure.

IO-Structure Diagnosis

The Web-IO Digital Logger replies to the *ReadDiagnosis* structure with a Diagnosis type structure.

Diagnosis (PC <- Web-IO)			
	WORD	send_sequenz	always 0
	WORD	rec_sequenz	always 0
	WORD	struct_type	0x00D0
	WORD	length	0x0018
	LONG	word_anz	in this release 3
	LONG	diag_error_count	number of current errors
	LONG	diag_errorbits0	binery error coding
	LONG	diag_errorbits1	binery error coding

The variable diag_error_count returns how many various errors are currently in the error memory. The Web-IO Digital Logger distinguishes a total of up to 64 different error states, whereby each set bit of the two variables diag_errrorbits0 and diag_errorbits1 stands for an error type.

The exact text description can be obtained on TCP Port 80. For a more detailed description, see the section on socket programming with command strings.

IO-Structure ClearDiagnosis

This structure clears the error memory of the Web-IO Digital Logger.

ClearDiagnosis (PC -> Web-IO)			
	WORD	send_sequenz	always 0
	WORD	rec_sequenz	always 0
	WORD	struct_type	0x00D2
	WORD	length	0x0008

IO-Structure Options

This structure is used to set certain options in the Web-IO Digital Logger. The *options* variable provides 32 bits for this.

Options (PC -> Web-IO)			
	WORD	send_sequenz	always 0
	WORD	rec_sequenz	always 0
	WORD	struct_type	0x01F0
	WORD	length	0x0010
	LONG	word_anz	in this release 1
	LONG	options	binary coding of options

In the current version of the Web-IO Digital Logger only Bit 0 in the *options* variable is used.

Bit 0 = 1 //the Web-IO Digital Logger returns the Structure register state when an output is set.
Bit 0 = 0 //the Web-IO Digital Logger does not reply when setting an output.

To ensure that the state after setting the output(s) is reliably returned, there should be a time of 150ms between two output-changing accesses.

5.5.9 IO structures for reading the logger contents

The data records stored in the Web-IO Digital Logger can also be read using binary structures.

IO structure LOGGER_REQUEST

In the *LOGGER_REQUEST* structure first the time separation for the desired data records is specified to the Web-IO Digital Logger.

LOGGER REQUEST (PC -> Web-IO)			
	WORD	send_sequenz	immer 0
	WORD	rec_sequenz	immer 0
	WORD	struct_type	0x01E0
	WORD	length	0x0014
	LONG	start_secs	Startzeit in sec ab 01.01.1970 00:00
	LONG	stop_secs	Stopzeit in sec ab 01.01.1970 00:00
	WORD	start_msecs	Startzeit ms
	WORD	stop_msecs	Stopzeit ms

The start and stop time are given in seconds starting with 01.01.1970 00:00:00 hours. In addition, the ms (in increments

of 100) for both times up to the next full second can be indicated.

IO structure LOGGER_ NOACCESS

The contents of the Data Logger can only be read by one instance at a time. If another read operation is active, the Web-IO Digital Logger replies with the *LOGGER_NOACCESS* structure.

LOGGER_NOACCESS (PC <-> Web-IO)			
	WORD	send_sequenz	immer 0
	WORD	rec_sequenz	immer 0
	WORD	struct_type	0x01E3
	WORD	length	0x000C
	LONG	logger_state	Status des Logger-Checks

In this case the connection should be ended and access to the Logger be retried at a later time.

IO structure LOGGER_CHECK

If access to the Logger is not already occupied, the structure *LOGGER_REQUEST* is answered by the Web-IO Digital Logger with the *LOGGER_CHECK* structure.

LOGGER_CHECK (PC <-> Web-IO)			
	WORD	send_sequenz	immer 0
	WORD	rec_sequenz	immer 0
	WORD	struct_type	0x01E1
	WORD	length	0x0014
	LONG	logger_state	Status des Logger-Checks
	LONG	memory_size	Zu durchsuchender Speicher
	LONG	memory_progress	Fortschritt der Suche

The Web-IO Digital Logger must first check how many data records are stored for the desired time period. This may take some time depending on how much memory is used and the size of the time period. To tell the application what the current status of the data record search is, the Web-IO Digital Logger returns the corresponding information using the *LOGGER_CHECK* structure.

There are 4 states for l*ogger_state*:

0 = No search (yet) active
1 = Initializing the search
2 = Loading a memory range to be searched
3 = Detecting the data records
4 = Search done

In *memory_size* the Web-IO Digital Logger returns the total size of the memory being searched.

The *memory_progress* parameter indicates how far along the search is.

As long as the return value for *logger_state* is not equal to 4, the application for its part must reply with the structure *LOGGER_CHECK*, whereby *logger_state*, *memory_size* and *memory_progress* are set to 0.

LOGGER_CHECK structures continue to be exchanged until *logger_state* has a value of 4.

IO structure LOGGER_DATASET

After the *LOGGER_CHECK* structure has been received with *logger_state* = 4, the data records can be requested.

This is done by sending an *EADriver* structure with structure type 0x01E2 from the application to the Web-IO Digital Logger.

LOGGER_DATASET (PC -> Web-IO)			
	WORD	send_sequenz	immer 0
	WORD	rec_sequenz	immer 0
	WORD	struct_type	0x01E2
	WORD	length	0x0008

The Web-IO Digital Logger then begins to send the data records. Each individual data record available for the time period is sent within a *LOGGER_DATASET* structure.

LOGGER_DATASET (PC <- Web-IO)			
	WORD	send_sequenz	immer 0
	WORD	rec_sequenz	immer 0
	WORD	struct_type	0x01E2
	WORD	length	0x0048
	LONG	maxcount	Gesamtanzahl der Datensätze
	LONG	processcount	Aktuell übertragener Datensatz
	LONG	time_secs	Zeitstempel sec ab 01.01.1970 00:00
	LONG	time_msecs	Zeitstempel ms
	WORD	outputs	Output Status
	WORD	inputs	Input Status
	LONG	reserved	nicht benutzt
	LONG	alarms	Alarmstatus
	LONG ARRAY[6]	counter	Counter 0 - 5
	LONG ARRAY[3]	errorcounter	Fehler für Bidirektionle Counter 1-3

The parameter *maxcount* indicates the total number of data records, whereas *processcount* indicates which data record is being currently sent.

Each data record is given the appropriate time stamp using *time_secs* (seconds beginning at 01.01.1970 00:00:00 hours) and *time_msecs* (milliseconds).

The fields *inputs* and *outputs* are used to send the binary status of the in- and outputs.

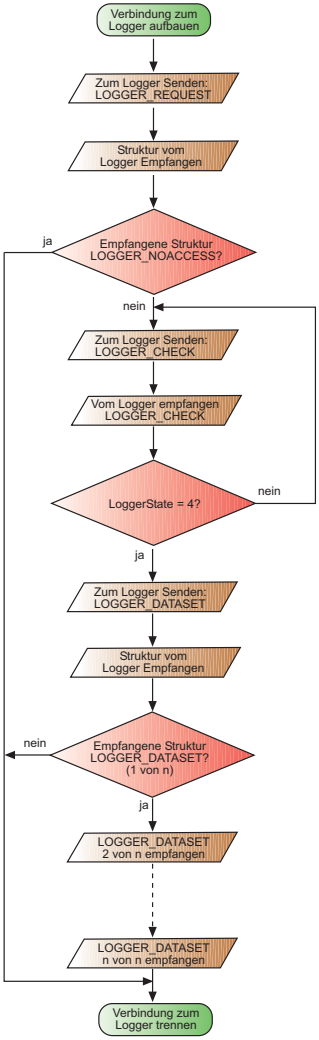
The alarm status is indicated in the first 12 bgits of *alarm*.

For the counter states of the counters the structure contains the 6x array *counter*.

If the bi-directional Counter mode is used but the direction could not be unambiguously determined, the count errors are incremented in the 3x array *errorcounter*.

The Web-IO Digital Logger sends all the data records for the selected time period without any new request from the application side. This procedure cannot be stopped.

The exact procedure for reading out the logger is shown again in the following flow chart.



5.5.10 IO structures in various languages

C / C++

The basic definition of structure is done in C using the command typedef.

```
typedef struct _SETBIT
{
    WORD send_sequence;
    WORD rec_sequence;
    WORD struct_typ;
    WORD length;
    WORD set_bits;
    WORD value;
    }SETBIT;
```

The *Variables* are in this case byte-aligned „Lowbyte First“ in memory.

To achieve byte alignment even for structures consisting of different variable types, please use the compiler option #pragma pack(1).

```
#pragmapack(1);

typedef struct _COUNTER
{
    word send_sequence;
    word rec_sequence;
    word struct_typ;
    word length;
    word    counter_index;
    long    counter_value
}   COUNTER;

#pragmapack()
```

Byte alignment in memory is important so that for example the sequence of bytes resulting from passing a pointer to the structure can be inserted directly into a TCP packet.

Visual Basic 5 and 6

Visual Basic offers the programmer the every easy-to-use Winsock control element for network communication via TCP/IP.

Unfortunately the VB Winsock control element does not allow working with structures. Instead, data communication is handled using string variables.

To program Web-IO Digital Logger applications in VB, you therefore need to represent the byte sequence as it is sent when sending a structure over the network in the form of a string.

The following source text excerpt shows how strings can be prepared for sending using the example of the ReadCounter structure.

First the necessary structures and variables need to be declared in an external module.

```
Public Type Struct_ReadCounter
  send_sequence As Integer
  rec_sequence As Integer
  struct_type As Integer
  length As Integer
  counter_index As Integer
EndType

Public Type Struct_Counter
  send_sequence As Integer
  rec_sequence As Integer
  struct_type As Integer
  length As Integer
  counter_index As Integer
  counter_value As Long
EndType

Public ReadCounter As Struct_ReadCounter
Public Counter As Struct_Counter
```

```
Public Str_ReadCounter As String * 10
PublicStr_CounterAsString*14
```

The following routine opens the connection to the Web-IO Digital Logger.

```
Private Sub ButtonConnect_Click()
'// Open TCP connection
  Winsock1.RemotePort = 49153
  Winsock1.RemoteHost = "172.16.232.17"
  Winsock1.Connect
End Sub
```

By invoking the next structure the *Read Counter* structure is written to a string and then send to the Web-IO Digital Logger.

```
Private Sub ButtonCounterRead_Click()
'// Fill the structure ReadCounter
  ReadCounter.send_sequence = 0
  ReadCounter.rec_sequence = 0
  ReadCounter.struct_type = &HB0
  ReadCounter.length = 10
  ReadCounter.counter_index = 11  '// Input-Counter 11
'// Sending of the individual bytes of the structure ReadCount
'// to the string Str_ReadCounter
'// always beginning with the Low byte of the individual INTEGER variables
'// followed by the High byte
'// or the higher value byte for LONG variables
  Str_ReadCounter = Chr(ReadCounter.send_sequence And &HFF) _
  + Chr((ReadCounter.send_sequence And &HFF00) / &H100) _
  + Chr(ReadCounter.rec_sequence And &HFF) _
  + Chr((ReadCounter.rec_sequence And &HFF00) / &H100) _
  + Chr(ReadCounter.struct_type And &HFF) _
  + Chr((ReadCounter.struct_type And &HFF00) / &H100) _
```

```
  + Chr(ReadCounter.length And &HFF) _
  + Chr((ReadCounter.length And &HFF00) / &H100) _
  + Chr(ReadCounter.counter_index And &HFF) _
  + Chr((ReadCounter.counter_index And &HFF00) / &H100)
  Winsock1.SendData (Str_ReadCounter)
EndSub
```

And finally the procedure that automatically ionvokes the Winsock control element when data are received by the network. The implementation here is exactly the reverse of that described above: The data arrive as a string and are written back to the *Counter* structure.

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
   '// Read arriving data into the string Str_Counter
   Winsock1.GetData Str_Counter
   '// Read out the individual bytes, calculate the value (Low Byte, High Byte)
   '// and save in the counter structure
   Counter.send_sequence = Asc(Mid(Str_Counter, 1, 1)) _
  + Asc(Mid(Str_Counter, 2, 1)) * &H100
   Counter.rec_sequence = Asc(Mid(Str_Counter, 3, 1)) _
  + Asc(Mid(Str_Counter, 4, 1)) * &H100
   Counter.struct_type = Asc(Mid(Str_Counter, 5, 1)) _
  + Asc(Mid(Str_Counter, 6, 1)) * &H100
   Counter.length = Asc(Mid(Str_Counter, 7, 1)) _
  + Asc(Mid(Str_Counter, 8, 1)) * &H100
   Counter.counter_index = Asc(Mid(Str_Counter, 9, 1)) _
  + Asc(Mid(Str_Counter, 10, 1)) * &H100
   Counter.counter_value = Asc(Mid(Str_Counter, 11, 1)) _
  + Asc(Mid(Str_Counter, 12, 1)) * &H100 _
  + Asc(Mid(Str_Counter, 13, 1)) * &H10000 _
  + Asc(Mid(Str_Counter, 14, 1)) * &H1000000
EndSub
```

The example shown above can of course also be applied to all other structures.

Delphi

Since Delphi offers very straightforward support of structures, using it to handle IO structures is a simple matter. Delphi also provides a socket control element for network communication that

can be used for exchanging data with the Web-IO Digital Logger.

The following source text excerpts show how you can use the *ReadCounter* and *Counter* structures to read the counter state of an input counter.

First the IO structures ReadCounter and Counter need to be defined as new variable types *RReadCounter* and *RCounter*.

```
RReadCounterr=packed record
    send_sequence:word;
    rec_sequence:word;
    struct_type:word;
    length:word;
     counter_index: word;
 end;

RCounter=packed record
    send_sequence:word;
    rec_sequence:word;
    struct_type:word;
     length:word;
     counter_index: word;
     counter_value: longword;
 end;
```

The structures must in any case be defined as *packed record*. The packed addition ensures that the sequence of Low and High byte is correct when sending (Low-Byte first).

The defined IO structures must be declared as variables:

```
  var
  Form1          : TForm1;
  ReadCounter    : RReadCounter;
  Counter           :RCounter;
```

The following procedure opens the TCP connection:

```
procedure TForm1.bConnectClick(Sender: TObject);
begin
  ClientSocket1.Host:= '172.16.232.17';
  ClientSocket1.Port:= 49153;
  ClientSocket1.Open;
end;
```

Invoking the next procedure sends the *Read Counter* structure to the Web-IO Digital Logger:

```
procedure TForm1.bSendClick(Sender: TObject);
begin
// Fill the ReadCounter structure
  ReadCounter.send_sequence = 0
  ReadCounter.rec_sequence = 0
  ReadCounter.struct_type = $B0
  ReadCounter.length = 10
  ReadCounter.counter_index = 11  '// Input-Counter 11
 ClientSocket1.Socket.SendBuf(ReadCounter,10);
end;
```

And finally the procedure that the Winsock control element automatically invokes when data are received by the network:

```
procedure TForm1.ClientSocket1Read(Sender: TObject;
  Socket: TCustomWinSocket);
var
  BytesReceived:Integer;
begin
  BytesReceived:=ClientSocket1.Socket.ReceiveLength;
  ClientSocket1.Socket.ReceiveBuf(Counter, BytesReceived);
end;
```

The above example can of course be applied to all other structures.

5.6 Box-to-Box

In this mode the inputs of a Web-IO Digital Logger are sent to the outputs of a second Web-IO Digital Logger and vice-versa.

In the case of Box-to-Box connections one Web-IO Digital Logger assumes the function of the master.

The second Web-IO Digital Logger functions as a slave. The slave waits for the master to connect with it.

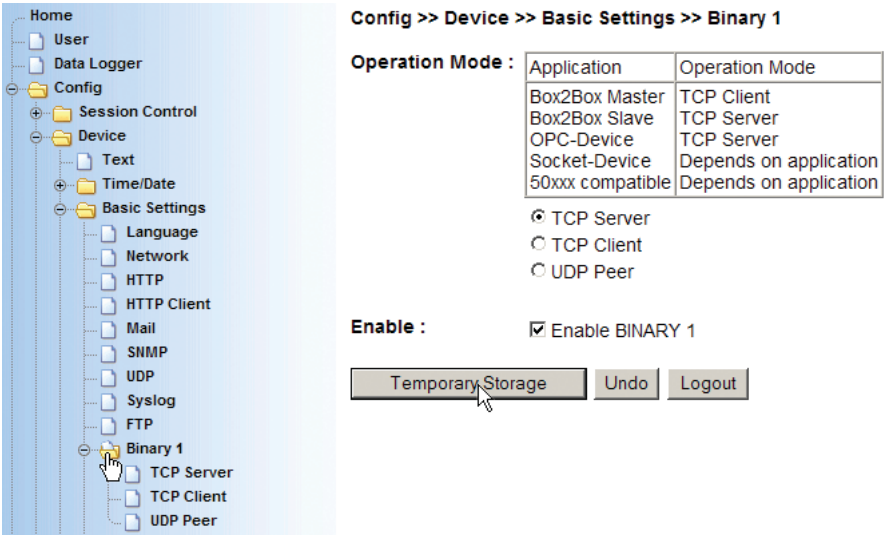
Both the master and the slave must be configured correspondingly.

For Web-IO Digital Loggers with relay outputs you must maintain a pause of at least 200ms between two switching operations on one output!

5.6.1 Configuring the Slave Web-IO Digital Logger

Required access rights: *Administrator*

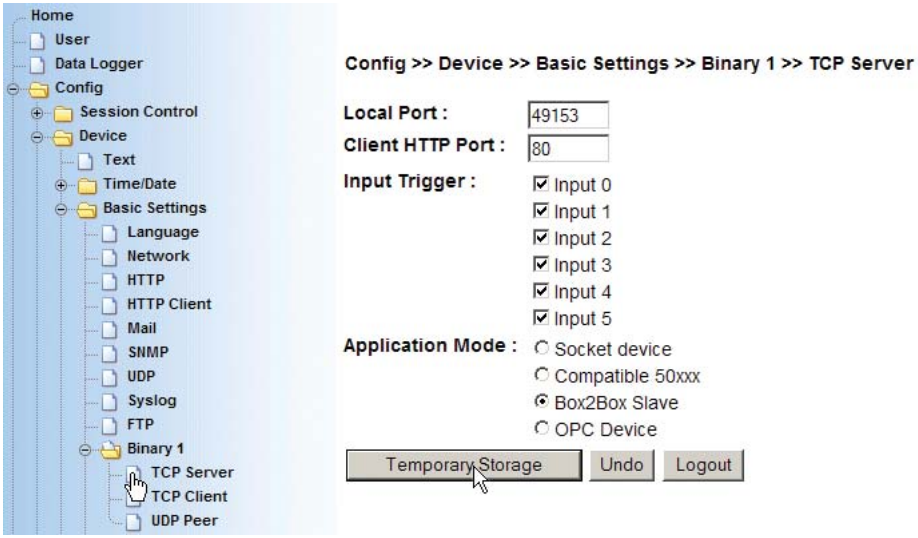
Select in the navigation tree des Slave Web-IO Digital Logger *Config >> Device >> Basic Settings >> Binary 1*



Set *TCP-Server* as the *Operation Mode*.

Then click on *Temporary Storage* to send the changes to the Web-IO Digital Logger.

Now select: *Config >> Device >> Basic Settings >> Binary1 >> TCP-Server.*



Local Port:
Unless your network administrator has given you other instructions, the factory default port 491 53 can be used.

A reason for changing the factory default Local Port might be for example a firewall that only allows certain port accesses.

In any case the Local Port for the slave must be identical with the Server Port *entry of the master.*

Client HTTP Port
Specifies the HTTP port on which the control connection to the Master box should be opened.

Unless otherwise specified, always use Port 80.

Input Trigger:
Here you activate the inputs that you want to set the corresponding outputs on the master.

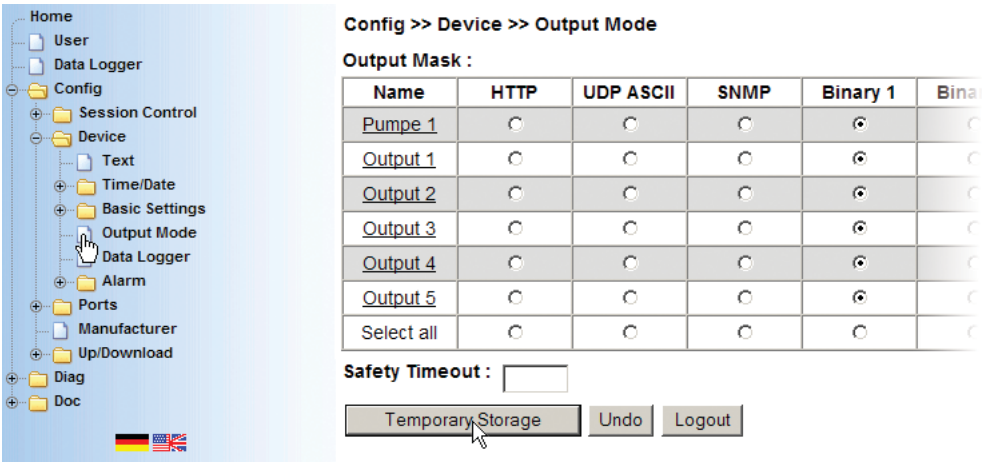
The Web-IO 1 2XDigital allows simultaneous access to the inputs from various operation modes.

This means for example that inputs 0-5 control outputs 0-5 on the master Web-IO Digital Logger, but can also be read via HTTP at the same time.

Application Mode
Select *Box2Box Slave*

After you have entered all the parameters, confirm by clicking on the *Temporary Storage* button.

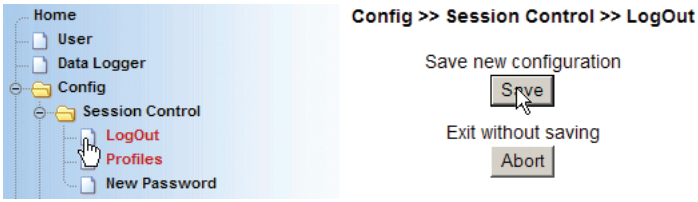
Select: *Config >> Device >> Output Mode*



Here you activate the outputs that will be set by the corresponding slave inputs for Binary 1 and confirm by clicking on *Temporary Storage.*

In contrast to the inputs, the outputs activated for Box-to-Box are no longer accessible for other operation modes.

Now you still need to activate the new settings. Use the *Logout* button or select *Config >> select >> Session Control >> LogOut.*



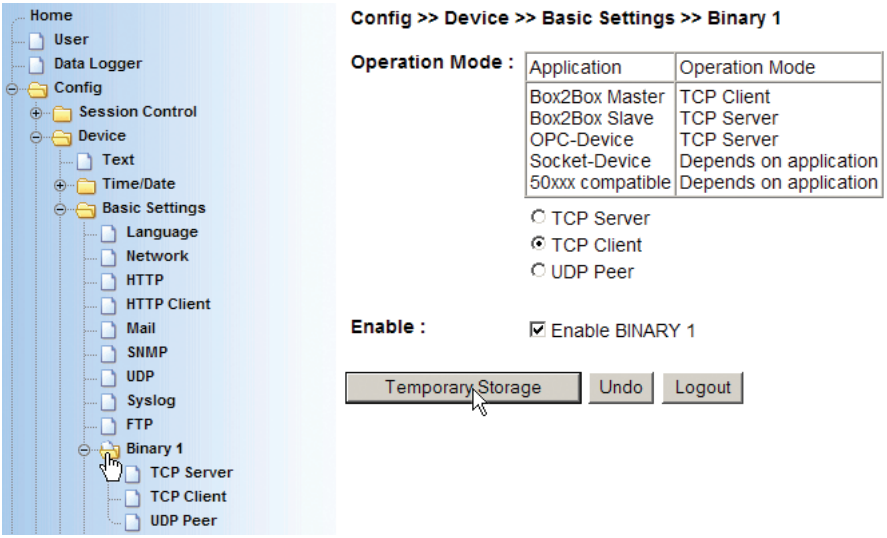
After clicking on *Save* all the settings are updated in the Web-IO Digital Logger and the start page is reopened in Default User mode. The master Web-IO Digital Logger can now connect to the slave Web-IO Digital Logger.

The Box-to-Box Master can only connect with the Slave if there is no Administrator or Operator login for *the Slave*

5.6.2 Configuring the master

Required access rights: *Administrator*

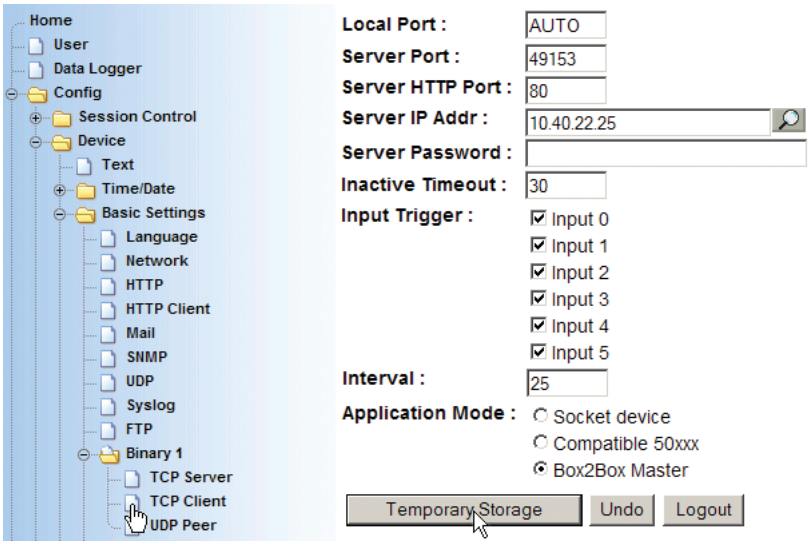
Select in the navigation tree: *Config >> Device >> Basic Settings >> Binary1*



Set *TCP-Client* as the *Operation Mode*.

Then click on the *Temporary Storage* button to send the changes to the Web-IO Digital Logger.

Select in the navigation tree: *Config >> Device >> Basic Settings >> Binary1 >> TCP-Client.*



The following parameters must be entered:

Local Port:
Unless your network administrator has given you other instructions, the factory default Port 49153 can be used.

ServerPort:
Here you must enter the *Local Port* set for the slave. Here again you can use the factory default 49153 if your network administrator has not specified otherwise.

Local Port and Slave Port do not necessarily have to be the same as set at the factory.

A reason for changing the factory default values for *Local Port* and *Slave Port* might be for example a firewall that only allows certain port accesses.

Server HTTP Port
Specifies the HTTP port on which the control connection to the Salve should be opened.

Unless otherwise specified, always use Port 80.

Server IP Addr:
Enter here the IP address of the Web-IO Digital Logger that you want to serve as a slave.

Server Password:
Here you enter the Administrator password for the slave Web-IO Digital Logger. If no password was assigned for the slave, leave this field blank.

Inactive Timeout
This parameter has no function in Box-to-Box mode, since a continuous connection is desired.

Input Trigger:
Activate here the inputs that will set the corresponding slave outputs.

The Web-IO 12XDigital allows simultaneous access to the inputs from various operation modes.

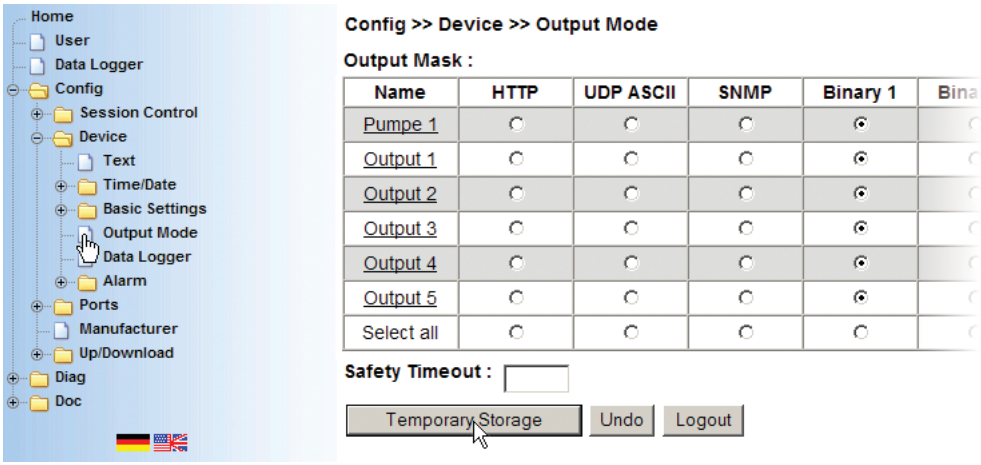
This means for example that inputs 0-5 control outputs 0-5 on the slave Web-IO Digital Logger, but can also be read via HTTP at the same time.

Interval:
If no interval is entered, the state of the inputs is sent to the outputs of the other Box-to-Box partner each time the state changes. Entering an interval sends the state cyclically even if there is no change.

If two locations are connected over a fee-based ISDN line, you should not use an interval, since depending on the timeout and interval the ISDN connection is never closed or is very frequently reopened.

Application Mode
Select *Box2Box Master*
After you have entered all the parameters, click on the *Temporary Storage* button.

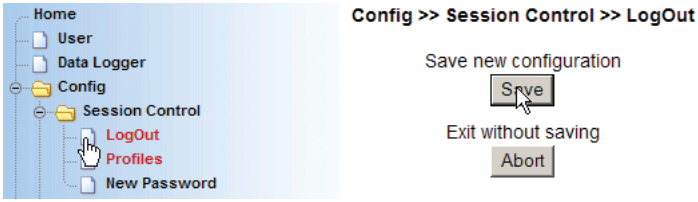
Select in the navigation tree: *Config >> Device >> Output Mode*



Here you activate the outputs that will be set by the corresponding slave inputs for Binary 1 and confirm by clicking on the *Temporary Storage* button.

In contrast to the inputs, the outputs activated for Box-to-Box are no longer accessible by other operation modes.

Now you must activate the new settings. Use the *Logout* button or *Config >> select >> Session Control >> LogOut.*



After clicking on the *Save* button all the settings are updated in the Web-IO Digital Logger and the start page is reopened in Default User mode. The master Web-IO Digital Logger then attempts to open a connection to the slave Web-IO Digital Logger.

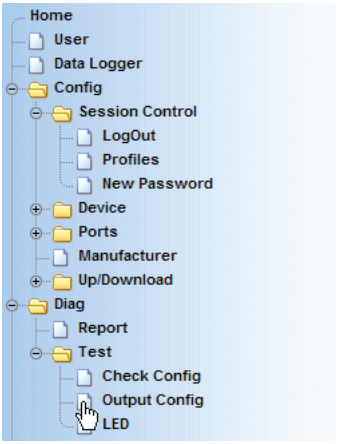
All the functions described here for Binary 1 can of course also be used under Binary 2. Thus a Web-IO Digital Logger A can be configured in the Binary 1 area so that the inputs/outputs 0-5

Box-to-Box work with a Web-IO Digital Logger B. In the Binary 2 area inputs/outputs 6 - 11 can then be configured so that they work Box-to-Box with another Web-IO Digital Logger.

5.6.3 GetBox-to-Box connection status

Required access rights: *Administrator*

The connection status of a Box-to-Box connection can be obtained from the navigation tree under *Diag >> Test >> Output Config*.



Output Konfiguration und Status

Output	Modus	Group Mode	Safety State
Pumpe 1	Binary 1	0 1 2 3 getrennt	unverändert
Output 1	Binary 1	0 1 2 3 getrennt	unverändert
Output 2	Binary 1	0 1 2 3 getrennt	unverändert
Output 3	Binary 1	0 1 2 3 getrennt	unverändert
Output 4	Binary 1	4 5 getrennt	unverändert
Output 5	Binary 1	4 5 getrennt	unverändert

last update: Do, KW12, 19.03.2009 16:36:16

Binary 1 Box2Box Slave TCP Client

connected to Box2Box Master!0.40.22.26 : 49153

Here you are shown which operation mode the individual outputs are working in. Additionally the current status of a Box-to-Box connection is displayed at the bottom of the Web page.

5.6.4 Stopping Box-to-Box mode

Box-to-Box Modus only by Master

Required access rights: *Administrator*

Stopping Box-to-Box mode should always be done using the corresponding configuration of the master. The master and

slave Web-IO Digital Logger must be connected on the network. Select in the navigation tree of the master : *Config >> Device >> Basic Settings >> Binary1 >> TCP Client* and delete the entry for *Server IP Addr*. Also set *Application Mode* to *Default.*

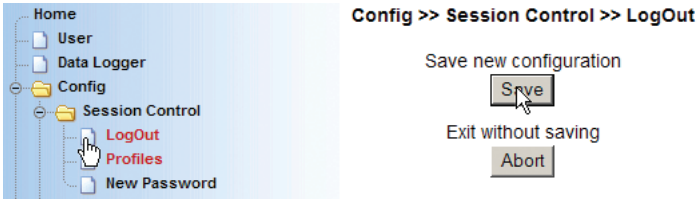
Confirm by clicking on the *Temporary Storage* button.

Then under *Config >> Device >> Basic Settings >> Binary1 >>* set the *Operation Mode* to *TCP Server .*

Confirm by clicking on the *Temporary Storage* button.

Now select in the navigation tree of the master : *Config >> Device >> Output Mode* and set the outputs that you no longer want to work under Box-to-Box to HTTP.

Confirm by clicking on the *Temporary Storage* button. Now you must still activate the new settings. Use the *Logout* button or in the Config menu *select >> Session Control >> LogOut.*



After clicking on the *Save button* all the settings are updated in the Web-IO Digital Logger and the start page is reopened in Default User mode.

Stopping Box-to-Box Modus for the Slave Web-IO Logger

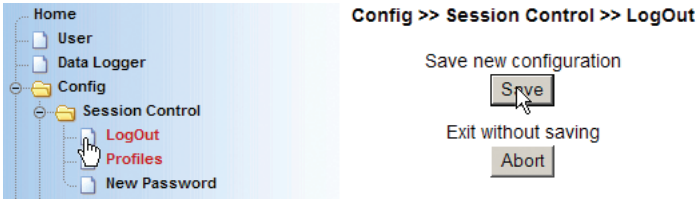
Required access rights: *Administrator*

Select in the navigation tree of the slave: *Config >> Device >> Basic Settings >> Binary1 >> TCP Server* and set *Application Mode* to *Default.*

Confirm by clicking on the *Temporary Storage* button.

Now select in the navigation tree *Config >> Device >> Output Mode* and set the outputs that you no longer want to work under Box-to-Box to HTTP.

Confirm by clicking on the *Temporary Storage* button. Now you must still activate the new settings. Use the *Logout* button or In the Config menu *select >> Session Control >> LogOut.*



After clicking on the *Save* button all the settings are updated in the Web-IO Digital Logger and the start page is reopened in Default User mode.

5.6.5 Stopping Box-to-Box Modus only for the Slave

If the master is no longer available, for example because there is no network connection but you still want to deactivate Box-to-Box mode for the slave, select in the navigation tree *Config >> Session Control >> LogOut*.

In the configuration frame there should be an additional button *Stop Box2Box Slave*.

If this button is not displayed, first click on the *Reset* button. This will restart the Web-IO Digital Logger.

After a new Login and invoking *Config >> Session Control >> LogOut* the *Stopp Box2Box Slave* button will be

displayed. Clicking on it will deactivate Box-to-Box mode in the slave.

5.6.6 Box-to-Box notes and limitations

Runtimes
Please note that when sending data using a network protocol, system-related runtime delays occur.

In contrast to a direct cable connection, whereby a switching signal is present directly on the actuator, latency times of between 20 and 50 ms must be factored in.

If active components such as routers, bridges or even ISDN segments are included in the transmission path, the latency time increases correspondingly, so that latency times of 500ms or more can occur.

Box-toBox with multiple partners
It is not possible to outputs signals from an input to the outputs of multiple Web-IO Digital Loggers. It is possible however to use Binary 1 and Binary 2 modes of a master Web-IO Digital Logger for example to link inputs 0 - 5 to Slave A by tinputs 6 - 11 to Slave B.

Mixing the various Web-IO Digital Logger models
It is possible to connect different Web-IO Digital Logger models with each other box-to-box. The condition is that the Web-IO Digital Logger is configured with a smaller number of ports than the Master.

1. Example:
You want to connect a Web-IO Digital Logger 2xDigital to a Web-IO 12xDigital Box-to-Box.

The Web-IO 12xDigital in this case is configured under Binary 1 as Box-to-Box Slave, whereby the input triggers for Input0 and Input1 are set. Also, in the Output Mode area the outputs 0 and 1 are enabled for Binary 1.

The configuration is then saved.

The Web-IO 2xDigital is configured as a normal Box-to-Box Master for Binary 1.

2. Example
You want to connect a 2 Web-IO 2xDigitals to a Web-IO 12xDigital Box-to-Box.

As in the first example, you first configure the Web-IO 12xDigital as a Slave. This is done both for Binary 1 and Binary 2. The input triggers 0 and 1 are set for Binary 1., and input triggers 2 and 3 for Binary 2. In addition, Outputs 0 and 1 are configured for Binary 1 and Outputs 2 and 3 for Binary 2 in the Output Mode area.

The configuration is then saved.

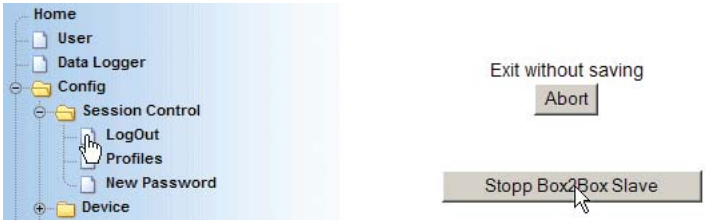
The 1st Web-IO 2xDigital is configured as a normal Box-to-Box Master for Binary 1. Proceed similarly for the 2nd Web-IO 2xDigital for Binary 2.

After saving the settings for all Web-IOs involved, Output 0 on the Web-IO 12xDigital follows changes on Input 0 of the 1st Web-IO 2xDigital. If Input 1 on the 2nd Web-IO 2xDigital changes, this results in a change on Output 2 of the Web-IO 12xDigital.

Old and new units in Box-to-Box combination
It is not possible to operate older 50xxx I/O Com-Servers with a Web-IO Digital Logger in Box-to-Box. Web-IO Digital Loggers working in Box-to-Box mode should have the same firmware version.

Problems with Box-to-Box
Especially when the IP addresses of a Box-to-Box pair have changed for example by a change in the infrastructure, it can happen that the Master can no longer open a connection to the Salve or vice-versa.

In such cases there should be an additional *Stopp Box2Box Slave* button under *Config >> Session Control >> LogOut*.



Clicking on this button resets any existing old connection parameters and opens the Slave box for a new connection.

5.7 OPC - standardized access

OPC (OLE for Process Control) is a software interface for accessing process data based on the Microsoft OLE technology.

Application programs such as visualization systems that use this interface are called OPC clients. On the other side of the interface are OPC servers. These are device drivers that represent a particular hardware in abstract form as a set of OPC variables.

The OPC server here implements the specifications of OPC Data Access 2.0 and controls devices in the W&T Web-IO Digital Logger product families, but also serial Com-Servers and the older digital I/O servers.

In terms of architecture we are dealing - in contrast to the usual implementation with a system service running in the background - with a monolithic application which contains the actual OPC server as well as operating elements for configuration and diagnostics.

5.7.1 Installing the OPC-Server

On our web site www.WuT.de, having selected the Englisch language version, on the left side you find „Insider search via article number". Enter here the article no. 57630, select „Tools" in the checkbox below and click „Go". On the page that will open, you select the link „OPC Server".

For ease of downloading the files have been compressed into a ZIP file. Save the decompressed files in any desired directory on your hard disk (e.g., C:/Temp). Start the setup program by for example going to *Run...* in the Start menu and then entering: „C:\Temp\Setup". This installs and registers the OPC server on your computer.

The OLE server name which the OPC clients will need later for connecting to the server is: *Wiesemann-Theis.DigitalIO.* The OPC server starts automatically in response to such requests. To configure the server you can also however run it manually. A corresponding entry *W&T OPC-Server* can be found in the Start menu under „Programs“.

Under Windows 95 the installation may fail with messages reporting missing functions in the OLE32.dll.

A more current version of this DLL which eliminates the problem can be found in the update package „DCOM for Windows 95“, which Microsoft provides free for downloading at:

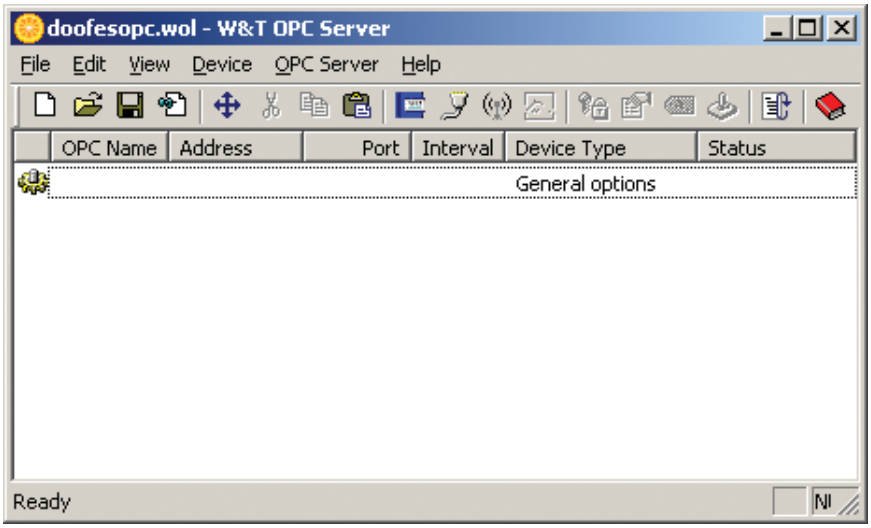
http://www.microsoft.com/com/dcom/dcom95/dcom1_3.asp.

5.7.2 Uninstalling

The OPC server can be removed from the Software component of the Control Panel. It will be found there in the list under *W&T OPC-Server for Digital-EA*.

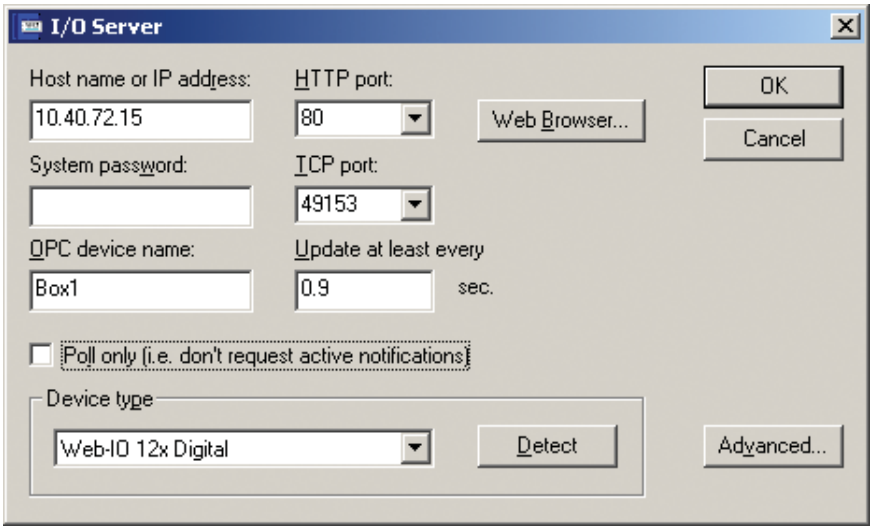
5.7.3 Configuration

First start the OPC server. For normal installation you will find the corresponding Start icon on your Windows interface under *Start >> Programs >> W&T OPC-Server for Digital-EA.*



Click on *New Device >> I/O.*

The following window will be opened:



Host name or IP address:
Enter here the IP address of your Web-IO Digital Logger. If a host name was also assigned by DNS or a similar name service, you may also use this name instead of the IP address.

TCP port number:
Must agree with the Local Port entry of the Web-IO Digital Logger (factory setting: 49153)

System password:
Enter here the Configurator or Administrator password for the Web-IO Digital Logger. If you are unsure of the correct password, please note after closing the window any status messages for the device in which problems are indicated („Password required“ or „Incorrect password“).

Consecutive device number:
This number is automatically assigned when adding a new I/O server or Com-Server and is intended to ensure that each device gets a different name (Box1, Box2, Com3, Box4, etc.). Manual changes may be desirable for example if you want to assign the same device number to the I/O ports and the serial port of an I/O server. The resulting device name would then be for example „Box1“ and „Com1“.

Refresh at least every ... sec.:
While the Web-IO Digital Logger is itself reporting changes to the inputs, the counter states for example have to be queried. This takes place during the interval specified here, whereby the entry is rounded off to a multiple of 0.1 seconds.

These regular queries also serve to detect a fault in the TCP connection to the terminal device as quickly as possible. A fault is presumed (and the connection reset) when no reply from the device has arrived between two queries.

A very small value at here can therefore make it impossible to open a connection.

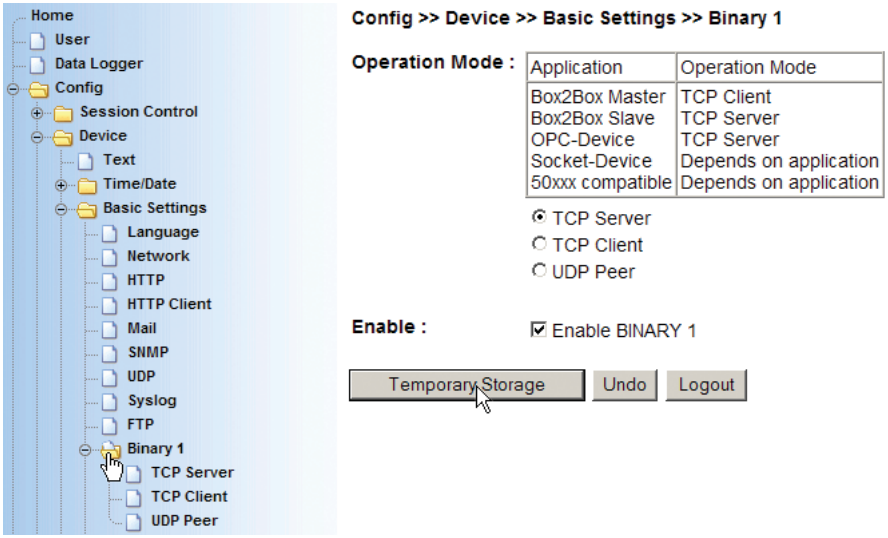
Poll only (do not subscribe to messages):
Opting out of automatic messaging has the advantage that the amount of data traffic on the network does not depend on any external events.

Device type:
Select Web-IO Digital Logger

5.7.4 Configuring the Web-IO Digital Logger as an OPC device

Required access rights: *Administrator*

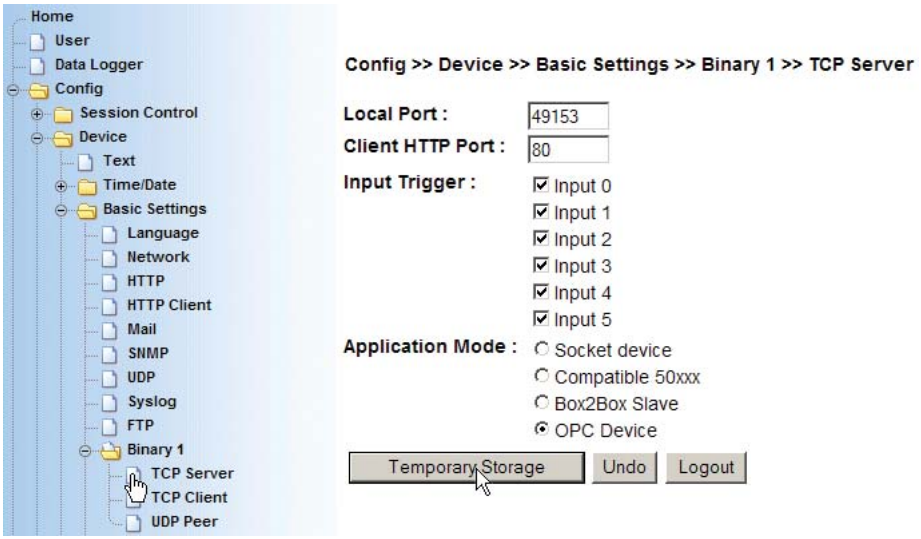
Select in the navigation tree of the Web-IO Digital Logger *Config >> Device >> Basic Settings >> Binary 1*



Set *Operation Mode* to *TCP-Server*.

Then click on the *Temporary Storage* button to send the changes to the Web-IO Digital Logger.

Select in the navigation tree: *Config >> Device >> Basic Settings >> Binary1 >> TCP-Server*.



Local Port:
Unless your network administrator has instructed otherwise, the factory default Port 491 53 can be used.

A reason for changing the factory default Local Port may be for example a firewall that only allows certain port accesses.

Client HTTP Port
Specifies the HTTP port on which the control connection to the OPC server should be opened.

Unless otherwise specified, always use Port 80.

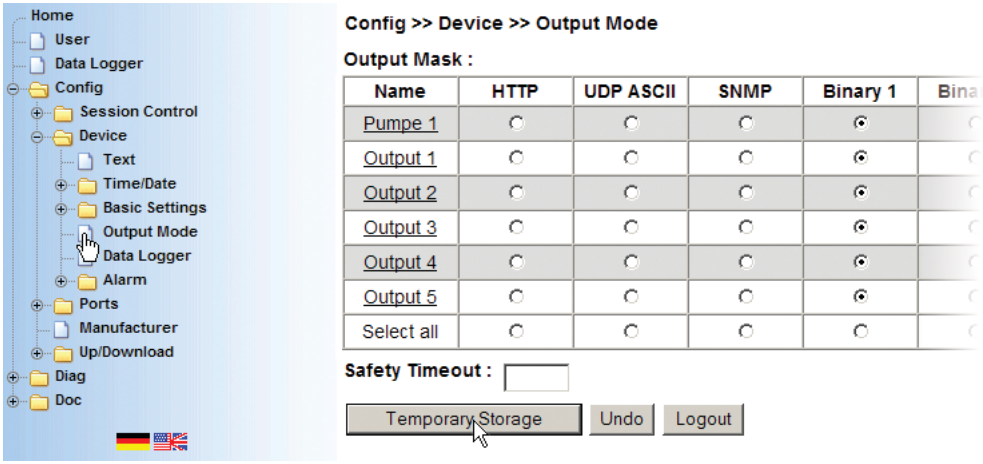
In any case the Local Ports for the Web-IO Digital Logger must be identical with the corresponding settings in the OPC server.

Input Trigger:
Activate here the inputs that will trigger a message to the OPC server when there is a state change.

Application Mode
Select *OPC Device*

After you have entered all the parameters, click on the *Temporary Storage* button.

Select in the navigation tree: *Config >> Device >> Output Mode*



Activate here the outputs to be controlled by the OPC server and confirm by clicking on the*Temporary Storage* button.

Now you must activate the new settings. Use the *Logout* button or in the Config menu *select >> Session Control >> LogOut.*

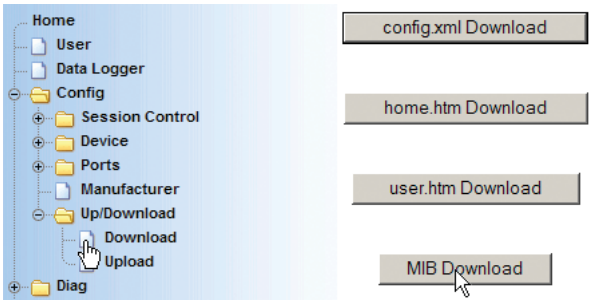
After clicking on the *Save* button all the settings are updated in the Web-IO Digital Logger and the start page is reopened in Default User mode. The Web-IO Digital Logger can now be accessed by the OPC server.

5.8 SNMP - In-/Outputs in management systems

The Web-IO Digital Logger can be incorporated into existing network managements systems by using SNMP protocol.

Both the total configuration as well as the controlling and monitoring of inputs, counters and outputs can be done using SNMP.

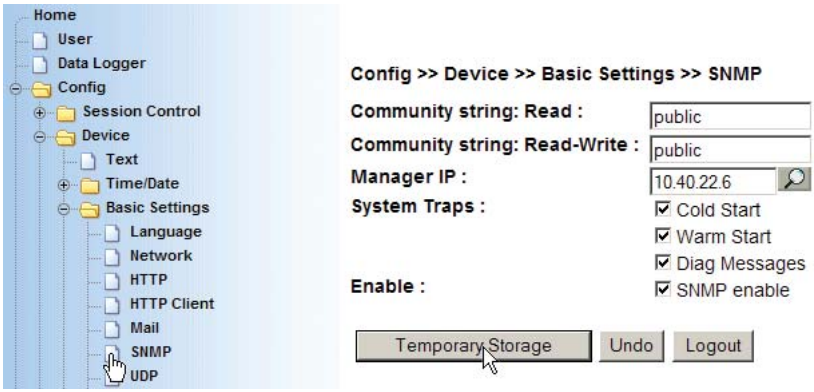
The private MIB can be directly downloaded from the unit.



5.8.1 Enabling SNMP access

Required access rights: *Administrator*

Select in the navigation tree des Web-IO Digital Logger *Config >> Device >> Basic Settings >> SNMP*



Community String: Read
Enter here the community string for SNMP access with read authorization.

Community String: Read-Write
Enter here the community string for SNMP access with read and write authorization.

Manager IP:
IP address of the SNMP Management-System.

System Traps:
The Web-IO Digital Logger offers the possibility of sending a system trap to the entered SNMP manager at a cold or warm start (Hardware-Reset and Power-ON or Software-Reset).

Select *Cold Start* or *Warm Start*.

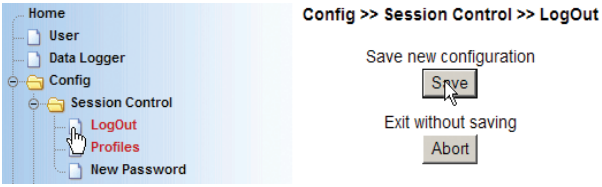
Diag Messages:
This option can be used to send error and diagnostic messages as an SNMP trap to the SNMP management system. As long as the internal memory of the Web-IO Digital Logger is not deleted, an error is sent as a trap only when it first occurs. Which errors are currently present can be seen by opening the Diag page in the browser.

See: http://<IP address of the Web-IO Digital Logger/diag

Enable:
At this point SNMP Enable must be activated in order to work with the Web-IO Digital Logger on SNMP level.

Confirm your entries by clicking on the *Temporary Storage* button.

If you have no more configuration changes to make, you still must activate the new settings. Use the *Logout* button or select *Config >> Session Control >> LogOut.*

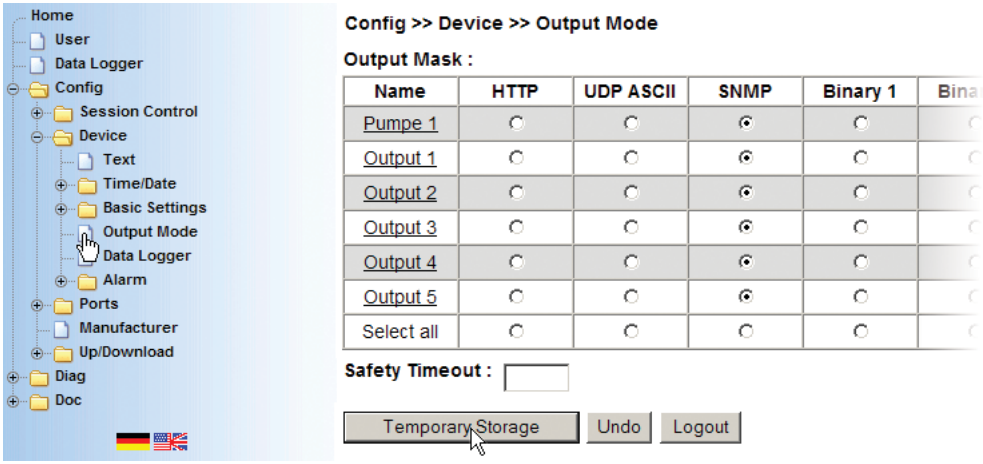


After clicking on the *Save* button all settings in the Web-IO Digital Logger are updated and the start page is restructured in Default User mode.

5.8.2 Activating the outputs for SNMP

Required access rights: *Administrator*

Select in the navigation tree of the Web-IO Digital Logger *Config >> Device >> Output Mode*



Activate here the outputs you want to control using SNMP protocol and confirm by clicking on the *Temporary Storage* button.

Now you must activate the new settings. Use the *Logout* button or In the Config menu *select >> Session Control >> LogOut.*

After clicking on the *Save* button all the settings are updated in the Web-IO Digital Logger and the start page is reopened in Default User mode.

5.8.3 Main steps for SNMP access

The private MIB provided by W&T can be used for all Web-IO Digital Logger devices. Separate OID branches are used for the various models

The procedure shown here applies as an example for the Web-IO 12xDigital Input, 12xDigital Output.

As soon as the SNMP access was enabled as described above, read access is possible at any time.

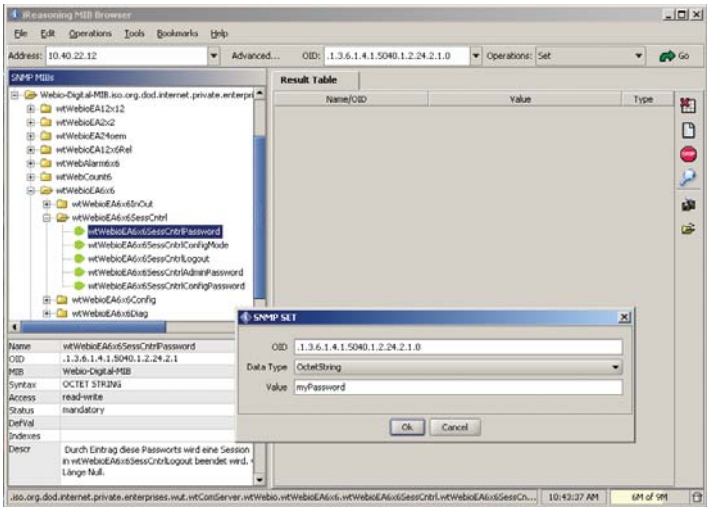
To have write access to the Web-IO Digital Logger, you must first open an SNMP session. This applies especially if you want to use SNMP for setting the outputs.

Session Login

To open a session, the adnministrator password must be entered into the variable

`wtWebioEA6x6SessCntrlPassword`

An open session remains active for 5 minutes and is then automatically cancelled.

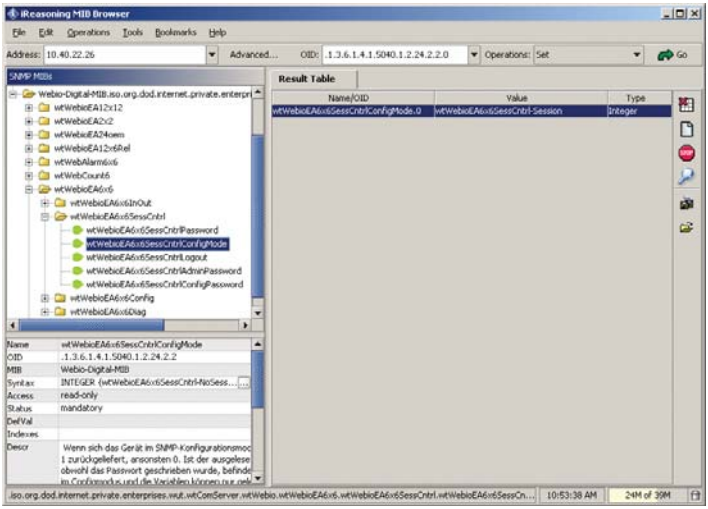


Checking a session for validity
By reading out the variables

`wtWebioEA6x6SessCntrlConfigMode`

you can check whether the session was successfully opened.

1 = Session opened, Web-IO Digital Logger in Configuration mode
0 = Opening of the session failed. Check whether the password was correctly entered.



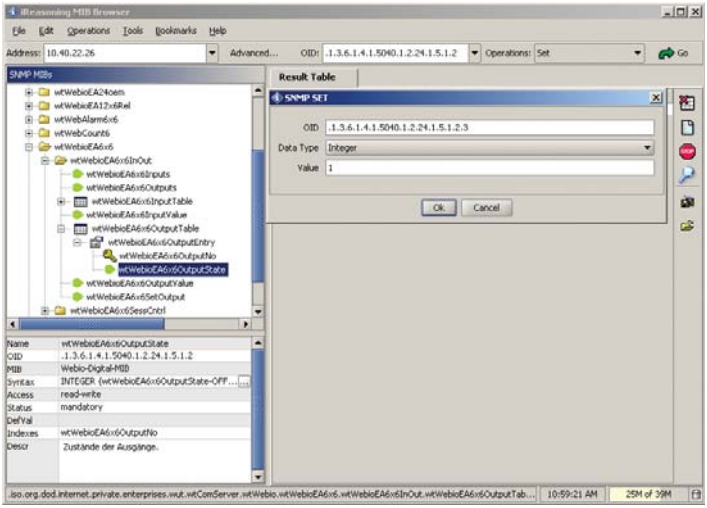
After successfully opening the session, the variables defined in the private MIB can be used to make any desired configuration changes or input and output accesses.

Changing an output

To change the state of individual outputs after a succesful session login, the SNMP variable

`wtWebioEA6x6OutputState`

is used. To determine which output to change, the number of the output must be apended to the corresponding OID, separated by a decimal point.



A value of 1 is sent for ON and a value of 0 for OFF. Here for example Output 3 is set to ON.

Change all outputs at once

In addition to setting individual outputs, you can change all the outputs at one time. To do this use the variable

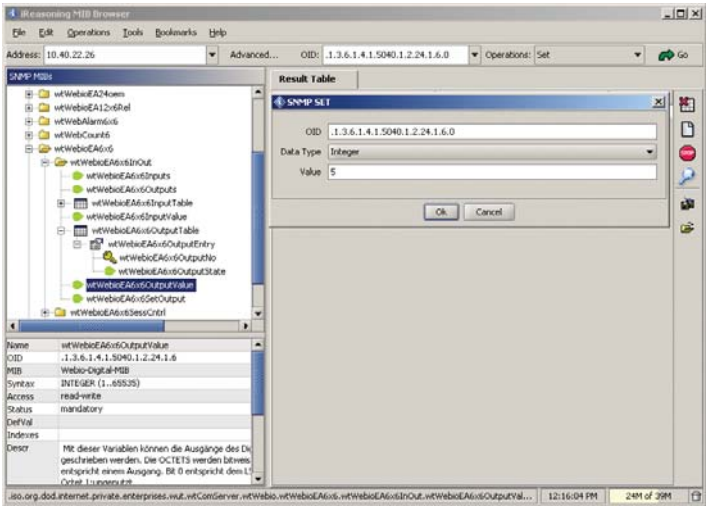
`wtWebioEA6x6OutputValue`

Transmitted is the integer value corresponding to the binary pattern of the outputs.

Example: You want to set Output 0 and Output 2 to ON. All other outputs should take on OFF status.

The transmitted value is calculated as follows:

$1x2^0 + 0x2^1 + 1x2^2 + 0x2^3 + 0x2^4 + 0x2^5 = 5$



Change certain outputs

Finally, the Web-IO Digital Logger offers the possibility of setting multiple outputs at a time while leaving others in their present state.

Here you work with an output mask and the output value.

The output mask determines which outputs will be changed. The output value, however, specifies which state these outputs should assume.

Example:
Set Output 1 to ON,
set Output 2 to OFF,
turn Output 4 ON,
turn Output 5 OFF

In other words you are changing outputs:

1,2,4 and 5.

The output mask is calculated as follows:

$0x2^0 + 1x2^1 + 1x2^2 + 0x2^3 + 1x2^4 + 1x2^5 = 54$

This corresponds to 0x36 hex.

To calculate the output value only the bit significances need to be added, which correspond to the ON state, i.e. 1.

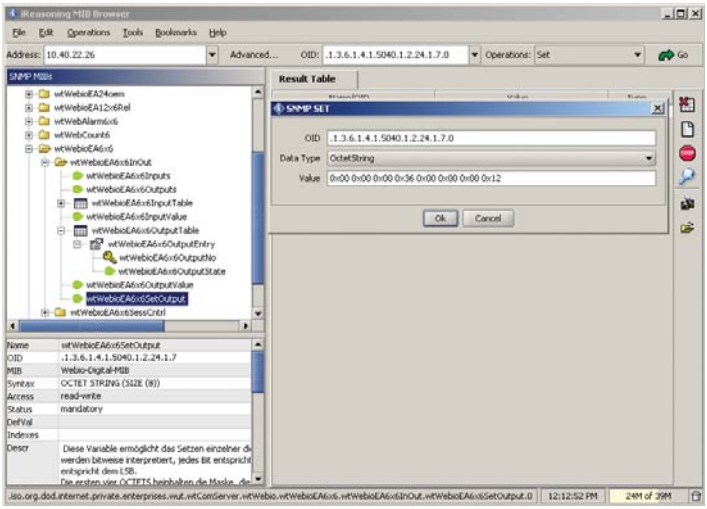
$1x2^1 + 1x2^4 = 17$

This corresponds to 0x12 hex.

The output mask and value are sent using the SNMP variable

`wtWebioEA6x6SetOutput`

as an 8-byte octet string.



Change configuration

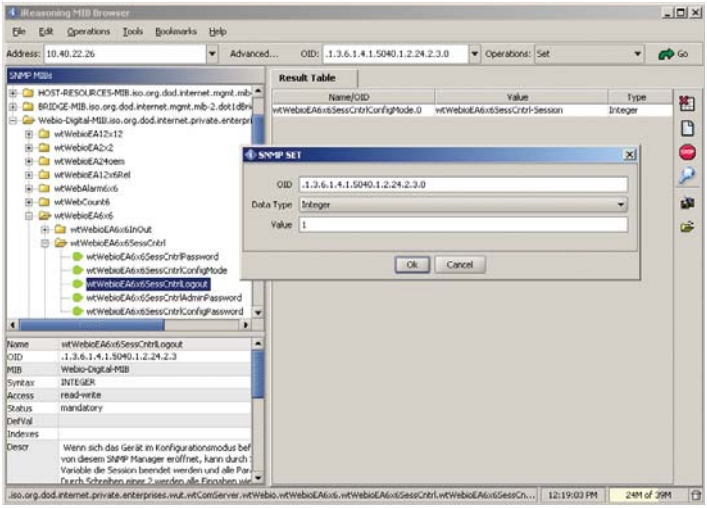
The base settings of the Web-IO Digital Logger can be read out using SNMP and even changed after a valid session login. The private MIB shows all the parameters which can also be changed from the browser interface. As opposed to setting outputs)done immediately in a valid session), the session must first be properly ended for the changes to become effective.

This is done by writing the variable

wtWebioEA6x6SessCntrlLogout

The value can be used to determine how the session is ended:

1 All changes are saved
2 Quit without saving



If no SNMP communication takes place during an open session for longer than 5 minutes, the Web-IO Digital Logger quits the session and all changes are cancelled.

Opening an SNMP session takes priority over an HTTP login. This means: A user with Config or Administrator rights loses his browser access as soon as an SNMP session is opened.

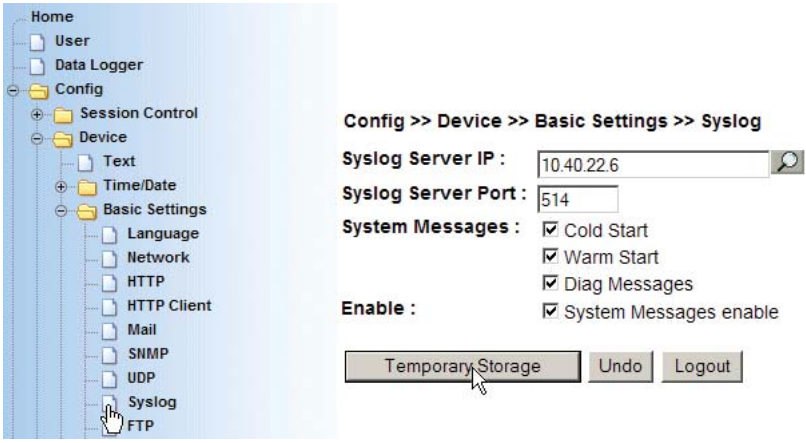
5.9 Syslog - Get system messages

The Web-IO Digital Logger can send system and error messages as well as alarm messages to a Syslog server using the Syslog protocol.

5.9.1 Enable Syslog

Required access rights: *Administrator*

From the Web-IO Digital Logger navigation tree select *Config >> Device >> Basic Settings >> Syslog*.



Syslog Server IP:
IP address of the Syslog server to which the messages should be sent.

Syslog Server Port
Enter here the port on which the Syslog server receives the Syslog messages. The default is Port 514. Allowed are all UDP ports between 1 and 65535 which are not otherwise used.

System Messages:
The Web-IO Digital Logger allows you to send a system message to the entered Syslog server at cold- or warm start

(hardware reset and Power-ON or software reset). The priority for these messages corresponds to the *Kernel.Info* status.

Select *Cold Start* or *Warm Start*.

Diag Messages:
This option allows you to send error and diagnostic messages to the Syslog server as a Syslog message with the priority *Deamon.Error*. As long as the Web-IO Digital Logger internal error memory is not cleared, an error is sent only the first time it appears. Which errors are currently in memory can be displayed using the Diag page in the browser.

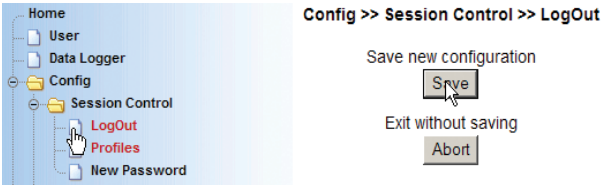
see: http://<IP address of Web-IO Digital Logger/diag

Enable:
At this point Syslog Enable must be selected in order to send Syslsog system messages using the Web-IO Digital Logger.

Additional Syslog system messages can be sent using the alarm functions of the Web-IO Digital Logger. Refer here to the section on *Alarms*.

Confirm your entries by clicking on the *Temporary Storage* button.

If you have no more configuration changes to make, you still must activate the new settings. Use the *Logout* button or select *Config >> Session Control >> LogOut.*



After clicking on the *Save* button all settings in the Web-IO Digital Logger are updated and the start page is restructured in Default User mode.

5.10 Alarms

In the Web-IO Digital Logger you can specify up to 12 different alarm conditions. Alarm messages can be issued depending on the status of the inputs and outputs. Four different network protocols are available:

- Mail (SMTP)
- SNMP
- Syslog
- UDP
- TCP (on an existing connection)
- TCP Client
- FTP Client

In addition to the alarm messages, the Web-IO Digital Logger also sends Alarm Clear messages when the alarm conditions are no longer met, i.e. the normal state is present again.

5.10.1 Determining the alarm conditions

Required access rights: *Administrator*

The alarm conditions can be specified in various trigger groups.

- Input Trigger
- Output Trigger
- System Trigger

The conditions for different trigger groups should not be mixed within an alarm.

Make triggering of an alarm dependent either on the status of the inputs or by one of the system triggers!

A distinction is made between two basic types of alarm triggers:

- Trigger with alarm state and normal state
- Trigger as single event

Trigger with alarm state and normal state
The trigger condition may be states ON or OFF for the in- or outputs, but may also be the system trigger Load Control.

If for example ON is selected as a trigger for an input, the alarm message is sent when this input goes to ON. When the input returns to OFF, the Web-IO Digital Logger sends the Alarm Clear message.


If an alarm condition returns to the normal state before the Web-IO Digital Logger was able to send the alarm message, only the Alarm Clear message is sent.

You could therefore say that these alarms are always finished with an Alarm Clear message.

Trigger as single event
Here triggers such as cold and warm start, timer interval and the ANY setting count for the inputs and outputs.

There is no normal state as such for these one-time events. SInce alarms shoudl always be finished with an Alarm Clear message, this type of trigger always generates only one Alarm Clear message.


There is no alarm message, but rather only Alarm Clear messages.

Alarm based on the status of the inputs and outputs
Select in the navigation tree *Config >> Device >> Alarm >> Alarm1* if for example you want to process the alarm conditions for Alarm 1.

Under *Input Trigger* and *Output Trigger* you can specify the following:

1. Which inputs or outputs to check for status
2. Which state (ON or OFF) meets the condition (AND operation)

The trigger conditions of the blocks Input Trigger, Output Trigger, Interval Timer and Load Control can be set independently (OR operation)

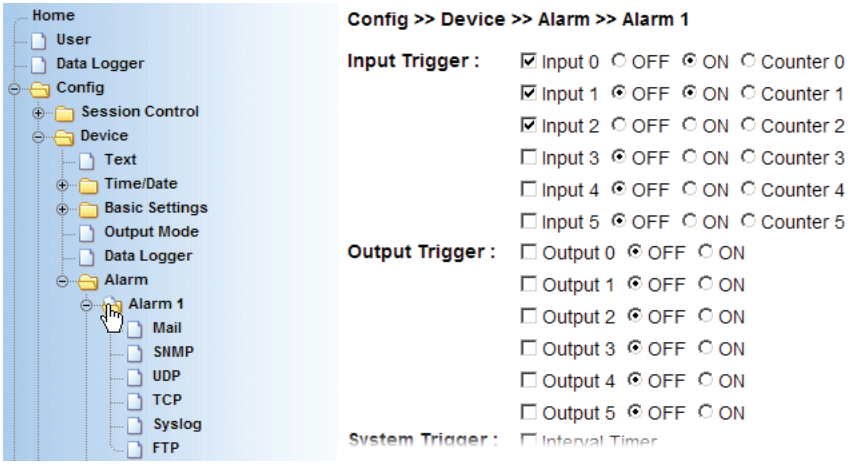
All inputs/outputs which are selected must meet the specified condition for the alarm to be triggered (AND operation).

If multiple inputs/outputs are configured with the ANY condition, these must switch at the same time for the alarm to be triggered.

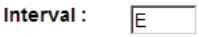
Alarms triggered using ANY only generate one Alarm Clear message!

Example:
Trigger an alarm when Input 0 and Input 1 are ON and no signal (=OFF) is on Input 1. Inputs 3 -11 are ignored.

The configuration would then look like this:



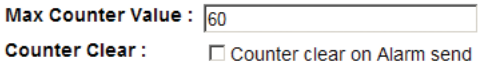
In addition you must specify whether an alarm is triggered only once when the alarm condition is met. Alternately you can specify an interval at which the alarm is repeated as long as the alarm condition persists.



In the Interval field you can specify either *E* for one time or an interval in minutes. Theoretical values between 1 and 4.2 billion minutes are possible. If the field is blank or 0, no alarm will be issued.

Alarm when a counter value is reached

Instead of responding to the state of the inputs and outputs, you can trigger an alarm when a counter state is reached. To do this you must enter a value between 1 and 4.2 billion in the Max Counter Value field. An alarm is issued when one of the activated inputs reaches this counter value.



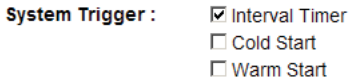
Counter Clear can be used to specify that the counter is reset to zero as soon as the specified counter state is reached.

Clearing a counter only takes place when the corresponding alarm has been sent. With rapid signal changes and slow alarm paths, such as e-mail, it can happen that the alarm triggers at 20, but the counter state is not cleared until 25 is reached.

For a time-accurate analysis of the counters, you should therefore use the Data Logger which can accurately keep absolute counter states to exactly 100ms.

Time-based alarm

Activating Interval Timer can allow the Web-IO Digital Logger to regularly issue alarms at intervals specified under *Interval* regardless of the input and output state.



In this way you can for example implement a keep-alive monitor.

At which times the alarm is triggered is determined in the area *Time Trigger*.

Time Trigger : Output of the alarm triggered by timer.

Field	Input [Number * , -]	Range of values
Minute	0-59/5	0-59
Hour	8-17	0-23 (0 is midnight)
Day_of_month	*	1-31
Month	*	1-12
Weekday	1-5	0-6 (0 is sunday)

The syntax for the time setting is derived from the Cron format used in Linux/Unix. Multiple entries separated by commas can be entered in one field.

The „*“ stands for any whole-number value. Example: * In the Minute field there is an alarm every minute, even if the conditions in the other fields are met.

Appending a „/“ to the actual value or value range allows you to specify a cycle value behind it. Example: 0-59/5 in the Minute field means there is an alarm every five minutes, even if the conditions in the other fields are met.

If you want to work with cyclical alarms, no wildcards „“ are allowed in front of the „/“.*

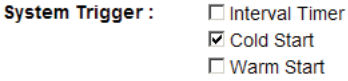
Using „-“ allows you to use time spans as triggers. Example: 8-17 in the Hour field means every hour between 8:00 a.m. and 5:00 p.m..

The above fields then mean an alarm is triggered every month on every Monday through Friday between 8:00 a.m. and 5:00 p.m. every five minutes.

It makes no sense to enter anything other than „E“ in the Interval *input field if you are not using* Time Trigger*!*

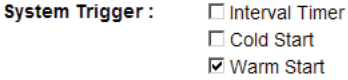
For timer-controlled alarms, the alarm is sent in the form of the Alarm Clear message

Alarm at cold start
Activating cold start causes the Web-IO Digital Logger to trigger an alarm after a Hardware or Power-ON Reset.



For cost start-controlled alarms, the alarm is sent in the form of the Alarm Clear message

Alarm at warm start
Activating warm start causes the Web-IO Digital Logger to send an alarm after a software reset, e.g., Connect to TCP Port 8888..



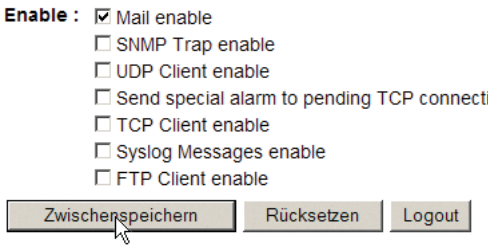
For warm start-controlled alarms, the alarm is sent in the form of the Alarm Clear message

After the alarm conditions have been specified, you can select which path to use for issuing the alarm.

5.10.2 E-Mail-Alarm - SMTP

To send an alarm via e-mail you must also activate *Mail Enable* (in addition to the alarm conditions) under *Config >> Device >> Alarm >> Alarm1*.

Required access rights: *Administrator*



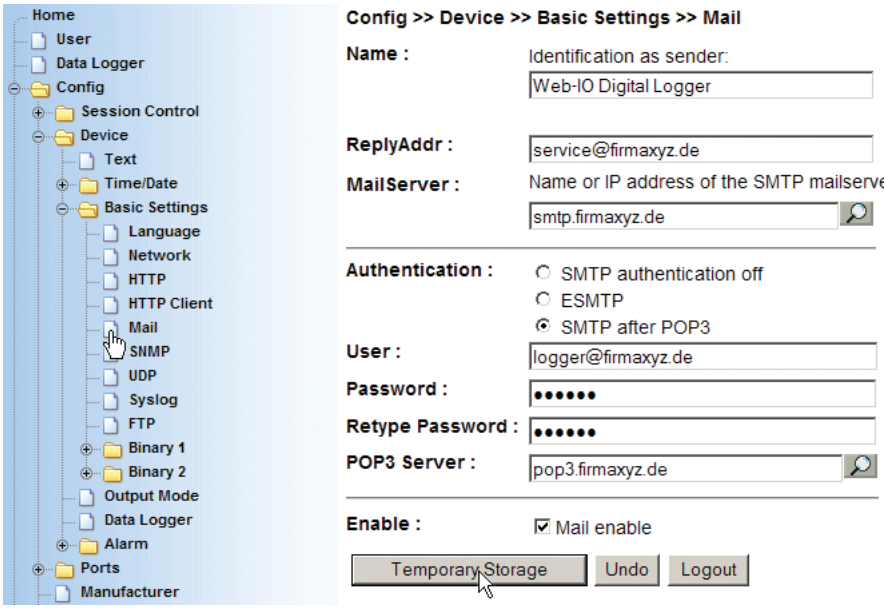
Send these settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

Basic settings for sending by e-mail

Required access rights: *Administrator*

In order for the Web-IO Digital Logger to send e-mail in the first place, you must make a few basic settings.

Go to *Config >> Device >> MailAdmin:*



Fill in the fields according to your application:

Name
Name entered as the sender when sending the e-mail.

Reply Address
E-mail address entered as the destination address in the outgoing e-mail when the recipient of an alarm e-mail selects *Reply*. SInce the Web-IO Digital Logger itself can only send but

not receive e-mail, you should enter here the e-mail address of the person who has technical responsibility for the Web-IO Digital Logger.

Mail Server
Here you must enter the host name or the IP address of the SMTP server used to send its alarm mails through the Web-IO Digital Logger.

Authentification
Here you can select whether you want to use e-mail authentication. depending on the ISP requirements, the Web-IO Digital Logger can be configured for a login with ESMTP or SMTP after POP3.

User
Enter here the user name for the e-mail account.

Password / Retype Password
Here the password assigned for the e-mail account must be entered.

POP3 Server
If you are using SMTP after POP3, you must enter here the name or the IP address of the POP3 server.

Enable
Set Mail enable to activate sending of e-mail.

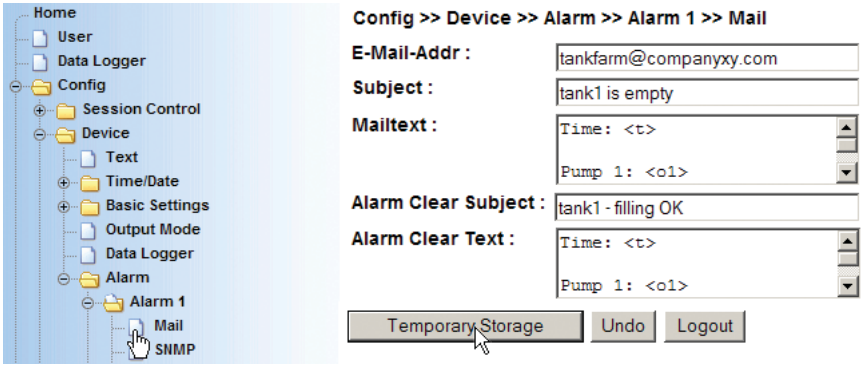
If the name of the SMTP or POP3 server is specified, a DNS server must be specified in the network settings.

Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

Specifying the e-mail recipient, subject and text

Required access rights: *Administrator*

Under *Config >> Device >> Alarm >> Alarm1 >> Mail* you can now specify who should receive an alarm mail, what should appear as the subject and what text the e-mail contains.



E-Mail Addr.
Enter the e-mail address for sending the e-mail when the alarm conditions are met. If you want multiple recipients for the e-mail, you must separate the e-mail addresses with colons, e.g. user1@domain.de; user2@domain.de; ...

Subject
Enter here the text to appear in the subject line of the e-mail.

Mailtext
This can be a variable text. In addition you can use alarm variables. These are placeholders for the state of the individual inputs and outputs as well as for the counter states.

Alarm Clear Subject
As soon as the configured alarm condition is no longer present, an Alarm Clear message is sent with the subject line entered here.

Alarm Clear Text
As with the mail text, here you may enter a freely formulated text for the Alarm Clear message.

In addition, alarm variables can be used in the subject line and in the mail text. These are placeholders for

status and name of of the individual inputs and outputs, as well as for the counter states of the counters. In addition, placeholders can be used for Date and Device Name.

Alarm Variable	Description
<dn>	Device Name look at: Config>>Device>>Text)
<i>	Input state as bit pattern in hex format
<i**x**>	State of Input No. **x** (ON/OFF),
<in**x**>	Name of Input No. **x**
<o>	Output state as bit pattern in hex format
<o**x**>	State of Output No. **x** (ON/OFF),
<on**x**>	Name of Output No. **x**
<c**x**>	Count state of counter no. **x** in decimal format
<t>	Time stamp with date and time formatted DD.MMM.YYYY hh:mm:ss
<$y>	Year formatted "YYYY"
<$m>	Month formatted "MM"
<$d>	Day formatted "DD"
<$h>	Hour formatted "hh"
<$i>	Minutes formatted "mm"
<$s>	Seconds formatted "ss"
	depending on the Web-IO Type **x** can be between 0 and 23

The example entered in the configuration frame would then look as follows as an e-mail:

Subject: tank 1 is empty

time: 18.Nov.2003 13:44:38

pump 1 = OFF

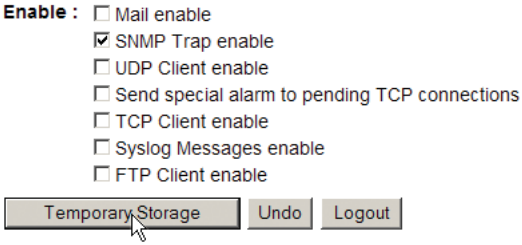
The mail tags were replaced by the values current at the time the mail was sent.

5.10.3 SNMP Alarm

In networks with a network management program you can also use SNMP protocol for sending alarms.

Under *Config >> Device >> Alarm >> Alarm1* you must also activate *SNMP Trap Enable* in addition to the alarm settings.

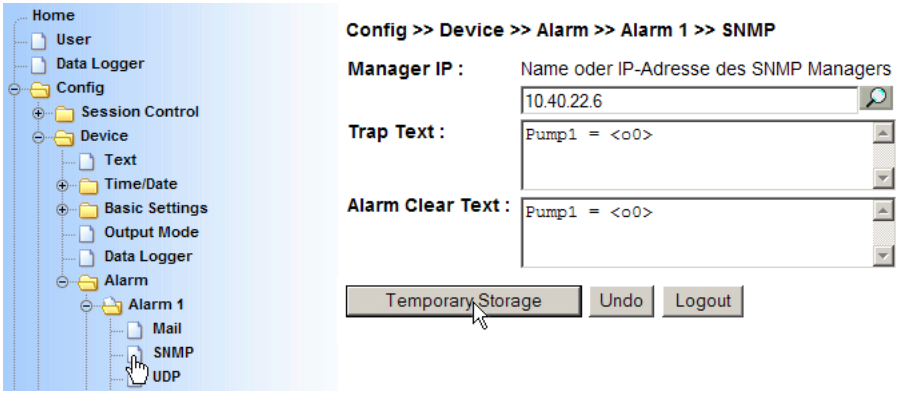
Required access rights: *Administrator*



Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

Select in the navigation tree *Config >> Device >> Alarm >> Alarm1 >> SNMP*

Required access rights: *Administrator*



Manager IP
Enter here the IP address of the network management system that will receive the SNMP information.

Trap-Text
Just as with the mail function, you can also enter a „trap text“ here which is sent to the manager software.

To incorporate the status of inputs and outputs as well as counter states into the text, you can likewise use alarm variables.

Alarm Clear Text
As in the case of the trap text, here you may enter here a freely formulated text for the Alarm Clear message.

The maximum length of the trap text must not exceed 450 bytes. Longer texts are cut off by the Web-IO Digital Logger.

Alarm Variable	Description
<dn>	Device Name look at: Config>>Device>>Text)
<i>	Input state as bit pattern in hex format
<i**x**>	State of Input No. **x** (ON/OFF),
<in**x**>	Name of Input No. **x**
<o>	Output state as bit pattern in hex format
<o**x**>	State of Output No. **x** (ON/OFF),
<on**x**>	Name of Output No. **x**
<c**x**>	Count state of counter no. **x** in decimal format
<t>	Time stamp with date and time formatted DD.MMM.YYYY hh:mm:ss
<$y>	Year formatted "YYYY"
<$m>	Month formatted "MM"
<$d>	Day formatted "DD"
<$h>	Hour formatted "hh"
<$i>	Minutes formatted "mm"
<$s>	Seconds formatted "ss"
	depending on the Web-IO Type **x** can be between 0 and 23

If you have entered an Administrator password, you will ave to enter this in your manager software as a „community string“!

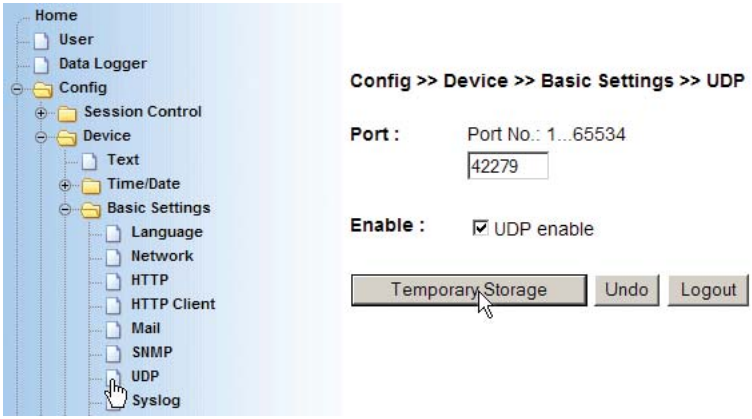
5.10.4 Alarm via UDP

Another way to pass an alarm to another network station is provided by UDP protocol. Here a corresponding program must be running on those PCs that are supposed to receive such an alarm.

To enable UDP for alarm sending, select *Config >> Basic Settings >> UDP* in the navigation tree.

Required access rights: *Administrator*

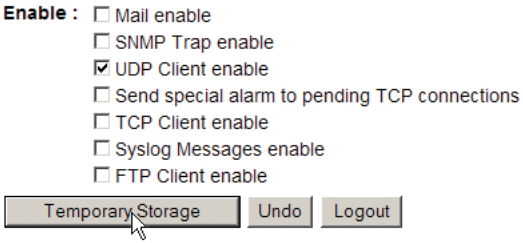
In the Port field enter the desired local port. The factory default setting is Port 42279. Check the UDP-enable box.



Send the selected settings to the Web-IO Digital Logger by clicking on the *Save* button.

To activate UDP for alarm transmission, select in the navigation tree *Config >> Device >> Alarm >> Alarm1* and select *UDP Client Enable* in the *Enable* area.

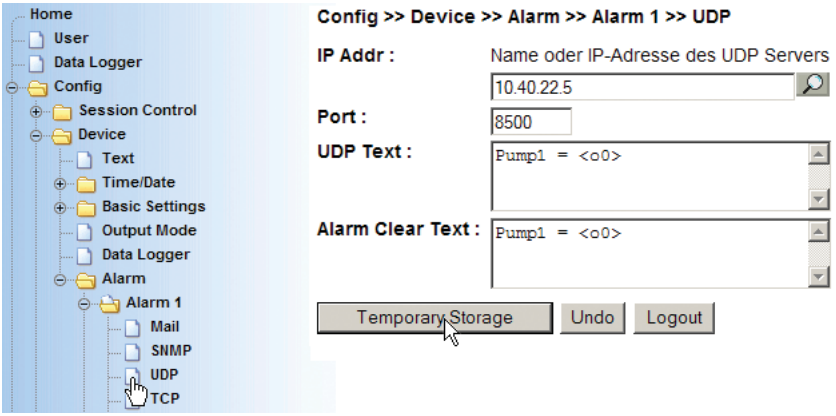
Required access rights: *Administrator*



Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

Under *Config >> Device >> Alarm >> Alarm1 >> UDP* the UDP-specific configurations can then be entered.

Required access rights: *Administrator*



IP Addr

Here you enter the IP address of the network station you want to receive the alarm message. Alternately you can enter a broadcast IP address to make the alarm messages accessible to all the participants in the network segment. IP-Adresse 172.16.232.255 for example would allow all network participants in segment 172.16.232.0 to receive the alarm.

Port
In this field you enter the local UDP port on which the application program should receive the UDP alarm messages.

UDP Text
Just as with the mail and SNMP function, you can enter here a text which is sent to the UDP application. To incorporate the state of inputs and outputs or counter states into the text, you can also use the e-mail tags.

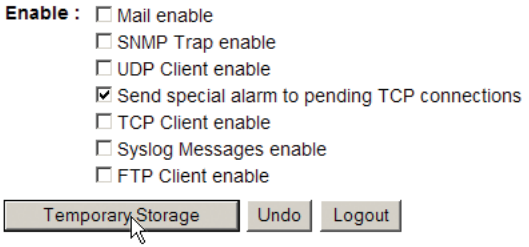
Alarm Clear Text
As in the case of UDP text, you can enter here a freely formulated text for the Alarm Clear message.

Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

5.10.5 Alarm message to your own TCP applications

In applications which communicate with the Web-IO Digital Logger over the socket interface via TCP, existing TCP connections can be used to send the input state when there is a state change.

So that a TCP data packet is sent to the active TCP client application when the alarm condition is met, go to *Config >> Device >> Alarm >> Alarm1* in the *Enable* area and select *Send input to pending TCP connections*



Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

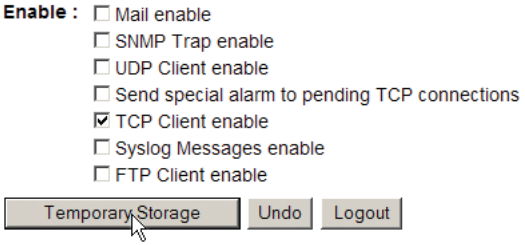
Once you have made all the entries pertaining to the alarm, activate the new settings using *Config >> Session Control >> LogOut*, *Save.*

5.10.6 Alarm message to a TCP server

You can also send an alarm messag to a TCP server. If there is an alarm the Web-IO Digital Logger opens a TCP connection to the configured Server and sends the message. The server hat to close the connection after recept of message.

To activate TCP client for alarm transmission, select in the navigation tree *Config >> Device >> Alarm >> Alarm1* and select *TCP Client Enable* in the *Enable* area.

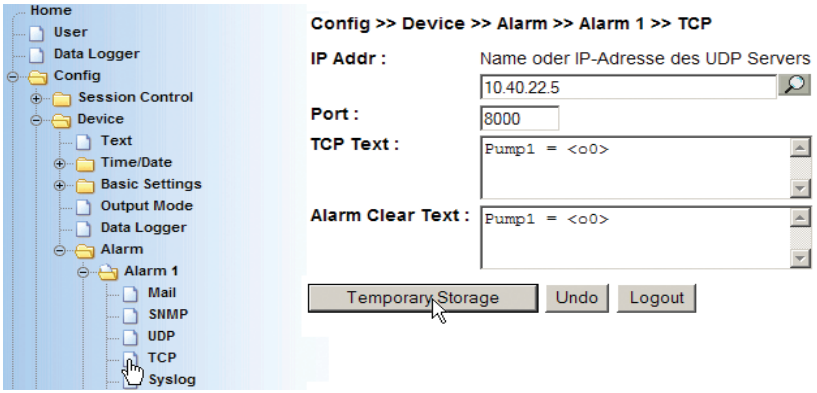
Required access rights: *Administrator*



Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

Under *Config >> Device >> Alarm >> Alarm1 >> TCP* the TCP-specific configurations can then be entered.

Required access rights: *Administrator*



IP Addr

Here you enter the IP address of the network station you want to receive the alarm message.

Port

In this field you enter the local TCP port on which the server application should receive the TCP alarm messages.

TCP Text

Just as with the mail and SNMP or UDP function, you can enter here a text which is sent to the TCP server application. To incorporate the state of inputs and outputs or counter states into the text, you can also use the Alarm tags.

Alarm Variable	Description
<dn>	Device Name look at: Config>>Device>>Text)
<i>	Input state as bit pattern in hex format
<i**x**>	State of Input No. **x** (ON/OFF),
<in**x**>	Name of Input No. **x**
<o>	Output state as bit pattern in hex format
<o**x**>	State of Output No. **x** (ON/OFF),
<on**x**>	Name of Output No. **x**
<c**x**>	Count state of counter no. **x** in decimal format
<t>	Time stamp with date and time formatted DD.MMM.YYYY hh:mm:ss
<$y>	Year formatted "YYYY"
<$m>	Month formatted "MM"
<$d>	Day formatted "DD"
<$h>	Hour formatted "hh"
<$i>	Minutes formatted "mm"
<$s>	Seconds formatted "ss"
	depending on the Web-IO Type **x** can be between 0 and 23

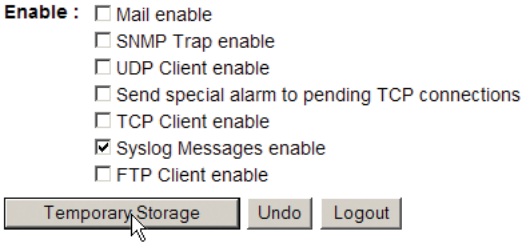
Alarm Clear Text
As in case of TCP text, you can enter here a freely formulated text for the Alarm Clear message.

Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

5.10.7 Sending alarm messages to a Syslog-Server

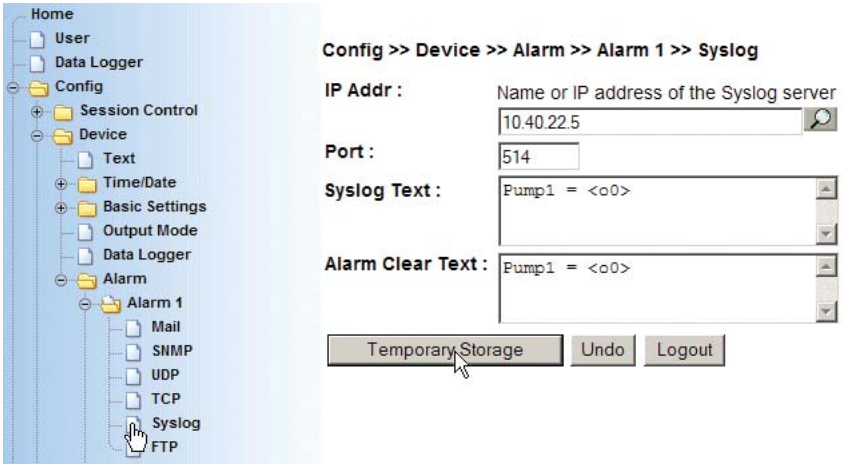
To send a system message to a Syslog server when the alarm condition is met, select *Syslog Message enable* under *Config >> Device >> Alarm >> Alarm1* in the *Enable* area.

Required access rights: *Administrator*



Click on the *Temporary Storage* button to send the settings to the Web-IO Digital Logger.b

Under *Config >> Device >> Alarm >> Alarm1 >>* Syslog you can then enter the Syslog-specific configurations.



IP-Adr
Here you enter the IP address of the Syslog server receiving the Syslog message.

Port
In this field enter the local UDP port on which the Syslog server should receive the Syslog message.

The standard port for Syslog is 514.

Syslog-Text
Just as with the mail and SNMP function, it is possible to enter a text here for sending to the Syslog server. To include the status of inputs and outputs or states of the counters you can also use the alarm variables.

Alarm Variable	Description
<dn>	Device Name look at: Config>>Device>>Text)
<i>	Input state as bit pattern in hex format
<ix>	State of Input No. x (ON/OFF),
<inx>	Name of Input No. x
<o>	Output state as bit pattern in hex format
<ox>	State of Output No. x (ON/OFF),
<onx>	Name of Output No. x
<cx>	Count state of counter no. x in decimal format
<t>	Time stamp with date and time formatted DD.MMM.YYYY hh:mm:ss
<$y>	Year formatted "YYYY"
<$m>	Month formatted "MM"
<$d>	Day formatted "DD"
<$h>	Hour formatted "hh"
<$i>	Minutes formatted "mm"
<$s>	Seconds formatted "ss"
	depending on the Web-IO Type x can be between 0 and 23

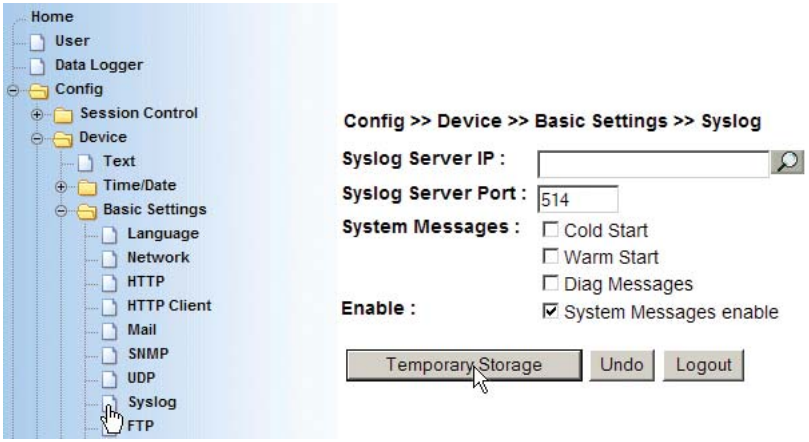
Alarm Clear Text

As in case of syslog text, you can enter here a freely formulated text for the Alarm Clear message.

Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

Finally you must enable the Syslog service.

To do this, select *Config >> Device >> Basic Settings >> Syslog.*

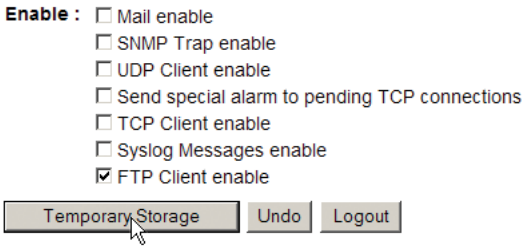


Syslog is enabled by checking *System Messages enable* and clicking on *Temporary Storage*.

5.10.8 Sending alarms via FTP Server

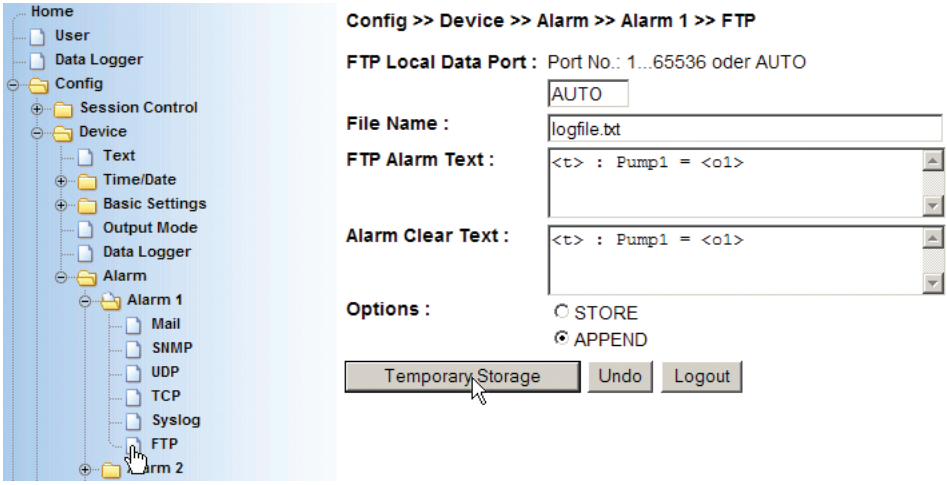
To write a message via FTP to a log file on an FTP server when an alarm condition occurs, highlight *FTP Client enable* in the *Enable* area under *Config >> Device >> Alarm >> Alarm1.*

Required access rights: *Administrator*



Send the settings to the Web-IO Digital Logger by clicking on the *Intermediate store* button.

Under *Config >> Device >> Alarm >> Alarm1 >> FTP* you can then enter the FTP-specific configurations.



FTP Local Data Port

Here you specify the local port for the connection on which the alarm messages are sent by the Web-IO Digital Logger to the FTP server.

If you specify AUTO, the Web-IO Digital Logger uses the next available port.

File Name

In this field you enter the name of the file in which the messages should be saved.

FTP Alarm Text

Just as with the mail and SNMP function, it is possible to enter a text here which is sent to the syslog server. To incorporate the status of inputs and outputs or the counter states of the counters, you can also use the alarm variables.

In addition, text variables can be used in the file name and in the FTP alarm. These are placeholders for status and name of of the individual inputs and outputs, as well as for the counter states of the counters. In addition, placeholders can be used for Date and Device Name.

Alarm Variable	Description
<dn>	Device Name look at: Config>>Device>>Text)
<i>	Input state as bit pattern in hex format
<i**x**>	State of Input No. **x** (ON/OFF),
<in**x**>	Name of Input No. **x**
<o>	Output state as bit pattern in hex format
<o**x**>	State of Output No. **x** (ON/OFF),
<on**x**>	Name of Output No. **x**
<c**x**>	Count state of counter no. **x** in decimal format
<t>	Time stamp with date and time formatted DD.MMM.YYYY hh:mm:ss
<$y>	Year formatted "YYYY"
<$m>	Month formatted "MM"
<$d>	Day formatted "DD"
<$h>	Hour formatted "hh"
<$i>	Minutes formatted "mm"
<$s>	Seconds formatted "ss"
	depending on the Web-IO Type **x** can be between 0 and 23

Alarm Clear Text

As in the case of FTP Alarm Text, here you can enter a freely formulated text for the Alarm Clear message.

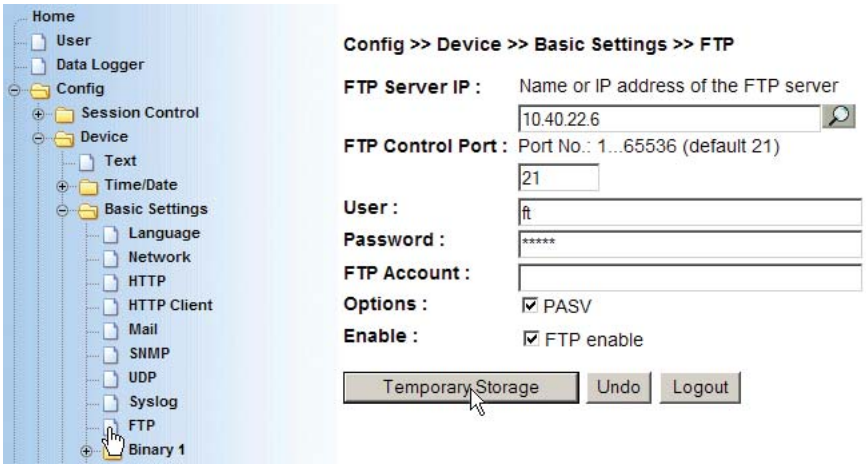
Option

Select *STORE* if you want the file to be completely overwritten each time a message arrives. For a file that expands with each message, select *APPEND*.

Send these settings to the Web-IO Digital Logger by clicking on the *Intermediate store* button

Finally, you must enable the FTP service and configure it.

To do this, use the path *Config >> Device >> Basic Settings >> FTP*.



FTP Server IP

Enter here the IP address of the FTP server on which you want to save the alarm messages.

FTP Control Port

Here the TCP port for the FTP control connection is specified. FTP servers normally use Port 21.

User

Here you enter the name of a registered user with write-access on the FTP server.

Password
User password, stored on the FTP server

FTP-Account
If applicable, enter here the account of the user.

Options
Specify here whether the FTP client should be active or passive. If unsure, consult with your system administrator.

Enable
Highlight *Enable* and click on *Intermediate store* to enable FTP as a service.

Once you have made all your alarm-related entries, activate the new settings with *Config >> Session Control >> LogOut, Save.*

5.10.9 General information about alarms

Multiple and even all send paths at the same time can be used for an alarm. For example, it is no problem to send both an e-mail and an SNMP-Trap.

6 General Functions

- Time / Date
- Poll watchdog - access monitoring
- Port configuration
- Changing the device-specific displays
- Upload / Download
- Diagnostics and testing

6.1 Time / Date - Setting

The Web-IO Digital Logger has an internal clock with date functions.

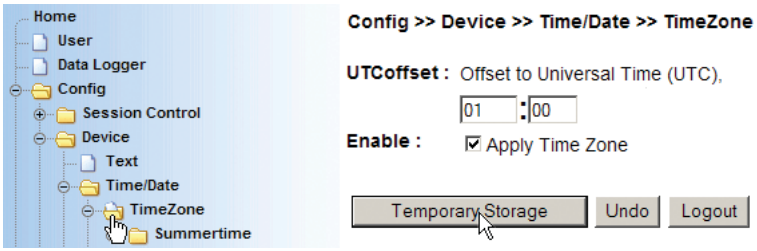
6.1.1 Time zones

Required access rights: *Administrator*

The Web-IO Digital Logger usesUTC time (universal time coordinated) internally, which is used as a general time base all over the World. The local time deviates from UTC depending on the time zone you are in. The time in Germany is UTC + 1h (in winter), the time in the United States is UTC - 5h, etc.

If the Web-IO Digital Logger will be used in another time zone, the offset can be adjusted accordingly.

To do this, go to *Config >> Device >> Time/Date >> TimeZone:*



UTCoffset
Enter here the difference between local and UTC time in the format hh.mm. A sign can also be used in the entry field („+“ or „-“).

Enable
To activate, check Apply Timezone to use the entered offset value.

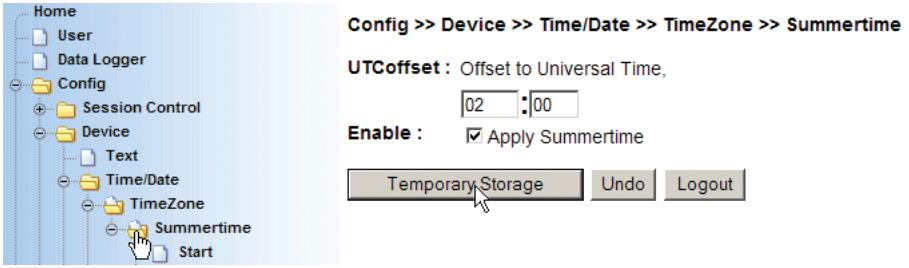
Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

6.1.2 Summer time / Winter time

Required access rights: *Administrator*

The Web-IO Digital Logger automatically distinguishes between summer and winter time.

To change the reference values for summer/winter time, select in the navigation tree *Config >> Device >> Time/Date >> TimeZone >> Summertime:*



UTCOffset
Enter here the difference between local summer time and UTC time in the format hh.mm. A sign can also be used in the entry field („+“ or „-“).

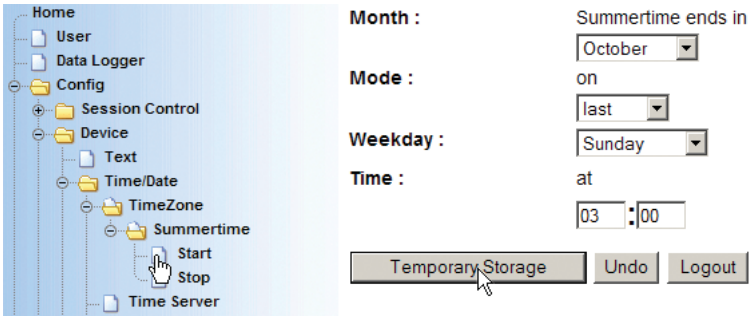
Enable
To activate, check *Apply Summertime* to use the entered offset value.

Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

6.1.3 Start and end of summer time

Required access rights: *Administrator*

To set when summer time begins, select *Config >> Device >> Time/Date >> TimeZone >> Summertime >> Start:*



Month
Select the month in which the change to summer time takes place.

Mode
If summer time does not begin on the last Sunday of the selected month, you can select a different mode here.

Weekday
If the time change does not take place for example from Saturday to Sunday, you can select a different day of the week.

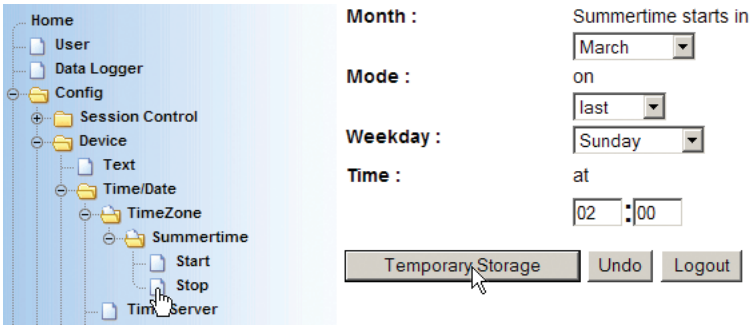
Time
Time at which the change takes place.

Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

6.1.4 End of summer time

m Required access rights: *Administrator*

To select when summer time ends, select *Config >> Device >> Time/Date >> TimeZone >> Summertime >> Stop:*



Month
Select the month in which summer time ends.

Mode
If summer time does not change on the last Sunday of the selected month, you can select a different mode here.

Weekday:
If the time change does not take place for example from Saturday to Sunday, you can select a different day of the week.

Time
Time at which the change takes place.

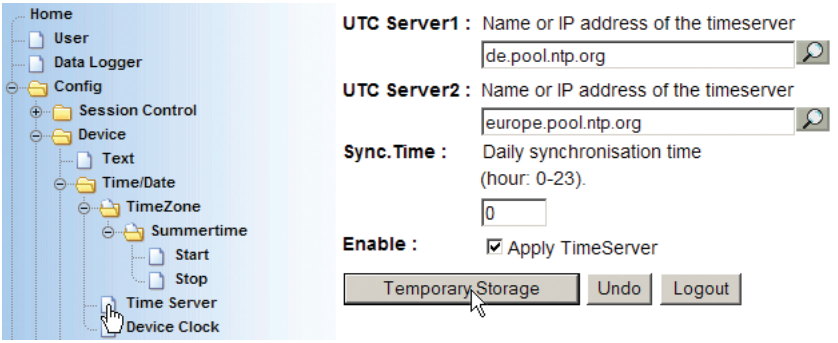
Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

6.1.5 Time-Server

Required access rights: *Administrator*

A time server can be used to synchronize the internal clock of the Web-IO Digital Logger. The Web-IO Digital Logger checks against the time server after it is powered up. There is also a daily time synchronisation at a selected time every day.

To enter the necessary settings for the time server, select in the navigation tree *Config >> Device >> Time/Date >> Time Server:*



UTC-Servers 1 + 2
Enter here one or two time servers that can be reached from the location of the Web-IO Digital Logger.

Sync Time
The hour the time will be synchronized with an external time server.

Synchronization with the time server takes also place approx. 30 seconds after the Web-IO Digital Logger is turned on or after a reset. If the Web page of the Web-IO Digital Logger is opened immediately after a new start, the old unsynchronized time will still be displayed. Only after opening the page again will the current time of day be shown.

Enable
Select *Apply Timeserver* so that the time of day will from now on be synchronized with an external time server.

Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

The factory default addresses can only be reached via an Internet connection. If names instead of IP addresses were specified for the time servers, a DNS server must be specified in the network configuration.

6.1.6 Manually setting time and Date

Required access rights: *Administrator* oder *Operator*

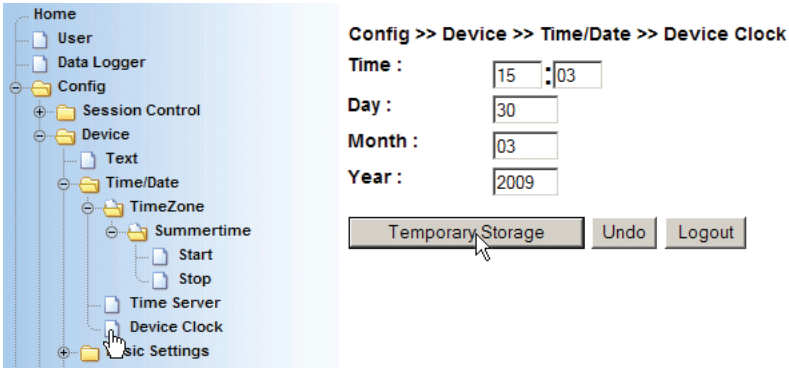
If there is no Internet access to a time server, the time of day can also be set manually under *Config >> Device >> Time/Date >> Device Clock.*

Time
Time in format hh:mm. It is not necessary to set to second accuracy.

Day
Format dd

Month
Format mm

Year
Format yyyy



Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

6.1.7 Activating all time/date settings

Once all the entries pertaining to the time and date have been made, activate the new setting under *Config >> Session Control >> Logout*, and then click the *Save* button.

If you log in again as a Configurator or Administrator immediately after activating the time server synchronization, it can happen that access will be denied to you after a few seconds.

This is because for security reasons the Web-IO Digital Logger resets the login rights after one hour. If for example the Web-IO Digital Logger has a time setting of 12:00 hours on 01.01.2002 (factory default at power on) just when the login takes place and the time server sets the time to 13:42 hours on 26.06.2003, the time span is considerably greater than one hour and the Web-IO Digital Logger cancels the access rights.

After logging in again the full hour can then be fully used up for configuration.

6.2 Safety Timeout / State - Access monitoring

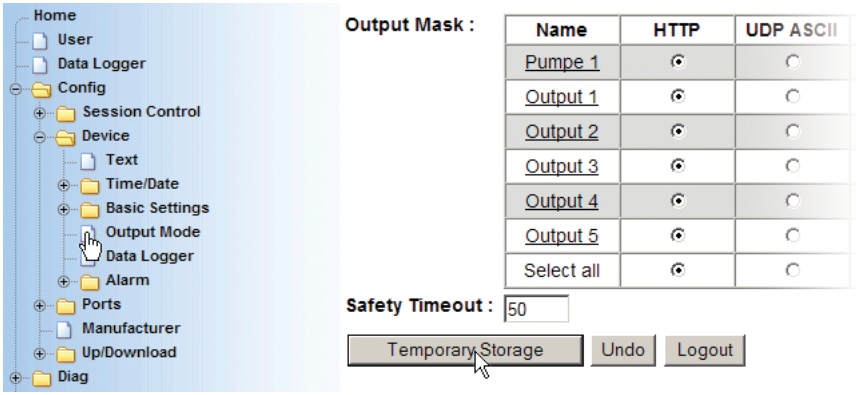
Components that communicate with each other over a network must rely on a functioning network infrastructure. Even though today's networks offer a high degree of transmission security, problems can never be entirely precluded.

To ensure predictable behavior of the Web-IO Digital Logger when there is a network disturbance, a *Safety Timeout* is provided. The associated watchdog monitors whether there has been a network access to the Web-IO Digital Logger within a configurable interval of time.The watchdog timer is reset each time there is a network access and plces the outputs in the configurable state when the set *Safety Timeout* time has been reached.

6.2.1 Specifying safety timeout

To configure the safety timeout, select in the navigation tree *Config >> Device >> Output Mode:*

Required access rights: *Administrator*



Safety Timeout
Enter here the time that should elapse before the absence of a network access causes the outputs to be set to the set pattern.

The entered value times 100ms is the timeout time; you can enter values between 1 and 9999.

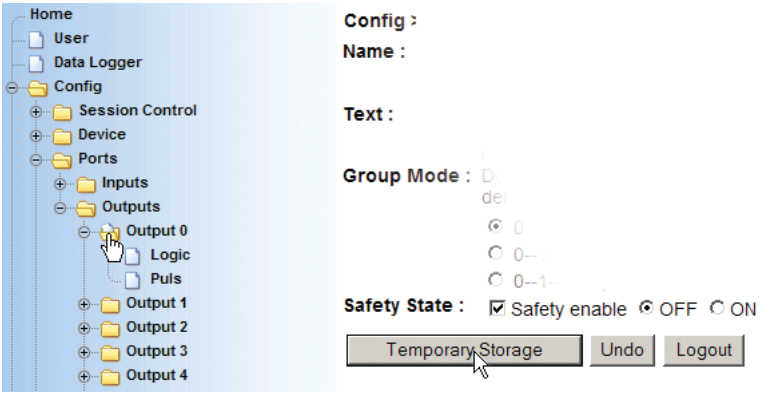
Example: a value of 50 represents 5 seconds.

As a rule of thumb, use a timeout that is 10x longer than the time between two scheduled or expected accesses.

Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

6.2.2 Specify Safety State

Select in the navigation tree *Config >> Ports >> Outputs >>* and for example *Output 0*



Safety State
Select *Safety State Enable* if you want the output to be set to a fixed state when there is a fault. Specify whether the output shall be switched to the *ON* or *OFF* state.

Outputs that are not selected keep the state that was present when the fault occurred.

All other settings on the page are not relevant to the *Safety State* and do not have to be changed.

Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

Once you have made all the settings, activate them under *Config >> Session Control >> LogOut*, *Save.*

6.3 Ports - Input- and output configuration

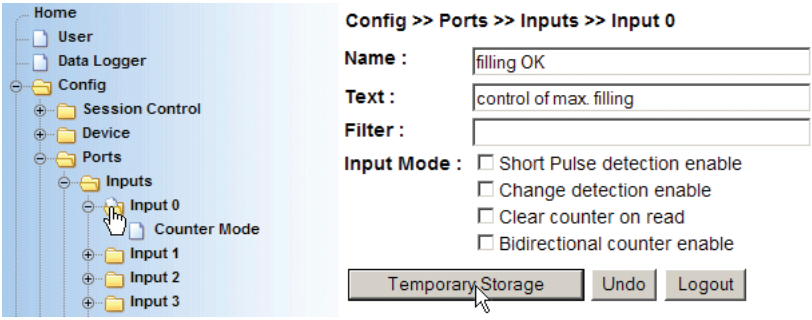
6.3.1 Input configuration

The inputs are organized into groups of 4. In addition to the galvanic isolation of the groups, Group 1 has the unique feature that the inputs can be selected for edge or pulse detection.

Individual base settings can be made for each input.

For example, to change the settings for Input 0, select in the navigation tree *Config >> Ports >> Inputs* and for Input 0 *>> Input 0*:

Required access rights: *Administrator* or *Config*



Name
Enter here a name for the input. This name will then be displayed in the browser for Input 0.

Text
Here you can provide a more detailed description, including for example the function of the input, which can then be viewed by clicking on the Input link in the browser.

Filter
Pulses with a duration shorter than specified here (duration in1/1000sec),are ignored.

Whenusing inputs that work with Short Pulse Detection, the configured filters have no effect

Input Mode: Short Pulse Detection Enable
This mode is available only for inputs 0 - 3 and also allows detection of extremely short pulses down to 50µs in length. In this mode only complete pulses, not edge transitions, can be counted.

If *Pulse Detection* is not enabled, signals must be present for at least 1ms in order to be detected and counted.

In both cases a maximum of 1000 counts per second can be detected.

Input Mode: Pulse Detection
Select this field if you want to configure a pulse count for the input.

This setting affects the way the counter counts.

If *Pulse Detection* is not activated each state change on the inputs is counted. A change from OFF to ON increments the counter just as a change from ON to OFF. So a complete puls will increment the counter by two. In this mode, has to be one state present for at least 1ms to be recorded.

If Pulse Detection is activated, there must be a change from ON to OFF and then a change from OFF to ON before the counter is incremented. The ON state must be present for at least 50ms for the pulse to be detected and counted.

In both cases, a maximum of 1000 state changes per second can be counted.

Input Mode: Change Detection Enable
In addition to counting complete pulses, here you can switch to counting edge changes.

This has no meaning if Short Puls Detection enable is active.

Input Mode: Clear counter on read
Here you can specify whether the counter is automatically set to 0 when it is read.

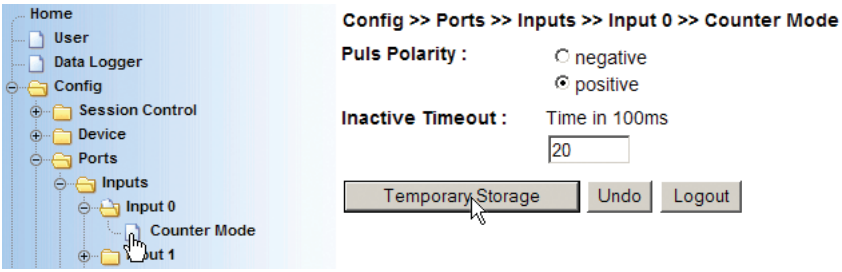
Input Mode: Bidirectional counter enable
Two adjacent inputs are combined for processing direction-dependent encoders.

Depending on which of the two inputs a valid edge is first detected, the corresponding counter is incremented by one.

Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

The necessary settings for direction-dependent counting are made under *Config >> Ports >> Inputs >> Input0 >> Counter Mode*.

Required access rights: *Administrator*



Pulse Polarity
Select here whether you want the Web-IO Digital Logger to trigger on positive or negative edges

Inactivity Timeout
If no valid edges are detected on either of the inputs within the set time, the counting is cancelled.

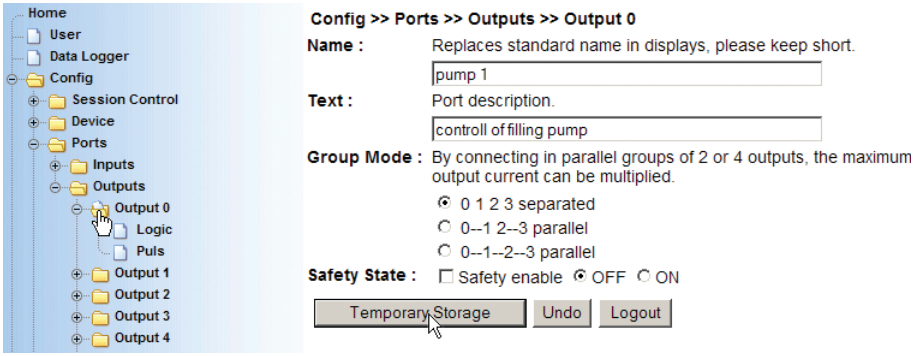
Send the settings to the Web-IO Digital Logger by clicking on the *Intermediate store* button.

Once you have made all the entries, activate the new settings under *Config >> Session Control >> LogOut, Save.*

6.3.2 Output configuration

To change the settings for Output 0 for example, select *Config >> Ports >> Outputs* and *>> Output 0:*

Required access rights: *Administrator* oder *Config*



Name
Enter here a name for the input. This name will then be displayed in the browser for Output 0.

Text
Here you can provide a more detailed description, including for example the function of the output, which can then be viewed by clicking on the Output link in the browser.

Group Mode
The outputs can be combined into groups of two or four to increase the maximum switching current. Combining two outputs allows you to switch a maximum current of 1A. With four outputs this increases to 2A. *(not applicable to Web-IO Digital Logger with relay outputs)*

Safety State
Select *Safety State Enable* if you want the output to be set to a fixed state when there is a fault. Specify whether the output shall be switched to the *ON* or *OFF* state.

This will only have an effect if a time is selected for *Safety Time-out* in *Config >> Device >> Output Mode*

A detailed description of *Safety State* can be found in the section *Safety Timeout / State.*

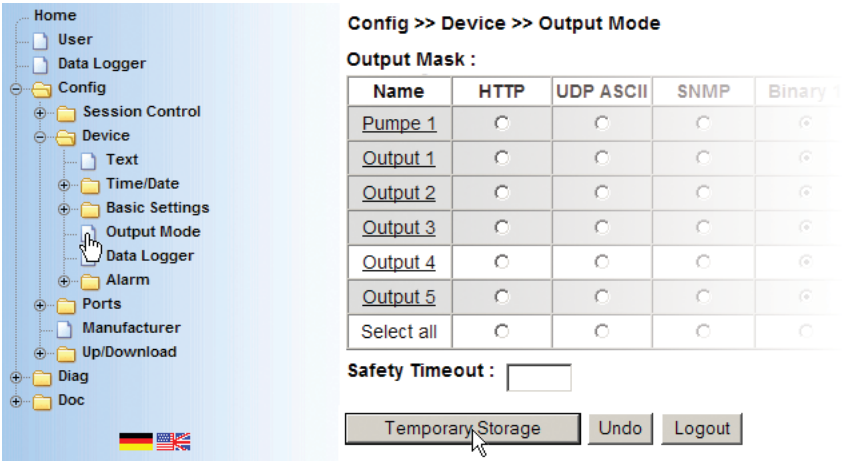
Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

Once you have made all the entries pertaining to the alarm, activate the new settings under *Config >> Session Control >> LogOut, Save.*

Outputs that are combined in a group should be configured for the same operation mode. Otherwise conflicts can arise if for example Outputs 0 and 1 are in Box-to-Box nmode and Outputs 2 and 3 can be switched from the browser, since each access switches all the outputs in a group at the same time.

An overview of the configured output groups can be found under *Config >> Device >> Output Mode.*

Required access rights: *Administrator*

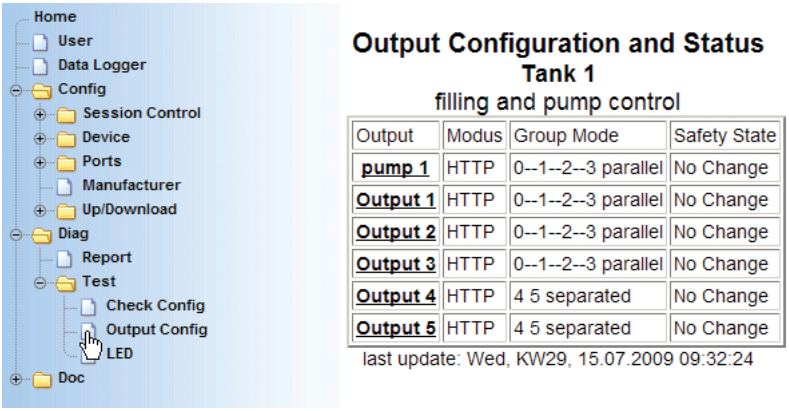


Here the outputs that are combined in a group are highlighted in the same color.

In the example shown, Outputs 0 - 3 represent a group. Outputs 4 and 5 and Outputs 6 and 7 represent two additional groups. Outputs 8,9,10 and 11 are ungrouped, and can be switched individually.

An additional way to see the groups at a glance is under *Diag >> Test >> Output Config* in the column *Group Mode.*

Required access rights: *Administrator*



6.3.3 Logic functions

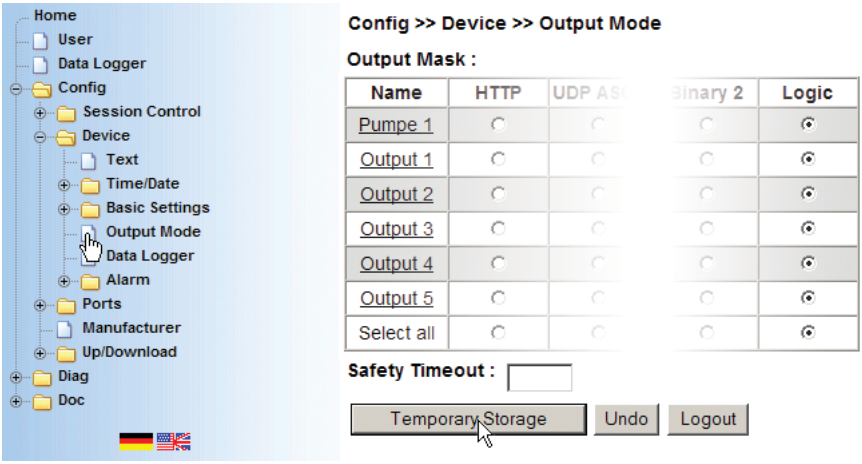
The Web-IO Digital Logger allows outputs to be switched depending on the state of certain inputs.

Example
Output 0 is set to ON when Input 0 and Input 1 are OFF.

For the logic functions to be applied to an output, the latter must first be configured to *Logic* mode.

To do this, use the path *Config >> Device >> Outputmode.*

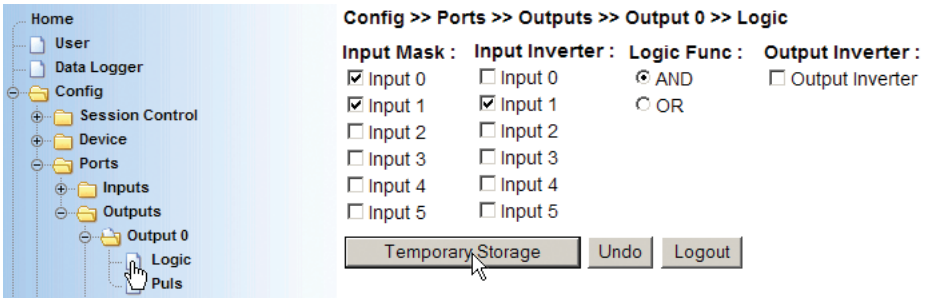
Required access rights: *Administrator*



Click on the *Logic* column for the outputs you want to control using the Logic function and then click on *Save*.

Select in the navigation tree *Config >> Ports >> Outputs >> Output 0 >> Logic:*

Required access rights: *Administrator* or *Config*



The example above looks like this in the browser:

Input Mask
Select here all the inputs you want to be included in the logic operation.

Input Inverter
Here you specify whether the state of the input should be inverted..

☐ = Input is evaluated as it is
☑ = The inverse of the input state is evvaluated

Logic Func
Specifies whether an AND or OR operation is applied.

Output Inverter
Here you specify whether fulfillment of the logic conditions sets the output to ON or OFF.

☐ = Output is set to ON
☑ = Output is set to OFF

Example
The configuration illustrated sets Output 0 to ON when Input 0 is ON and Input 1 is OFF.

Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

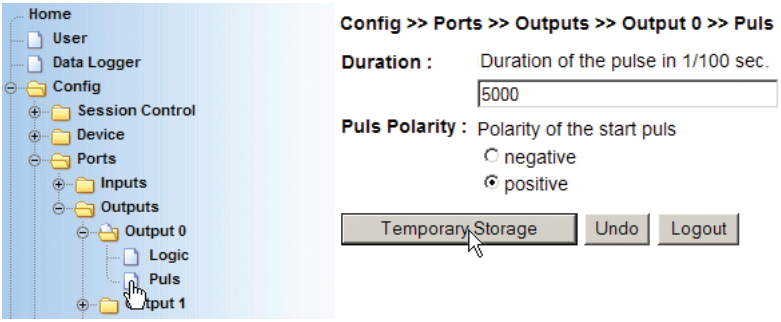
Once you have entered the conditions for all the outputs, activate the new settings under *Config >> Session Control >> LogOut, Save.*

6.3.4 Pulse mode

In addition to purely static switching of the outputs to ON or OFF, the Web-IO Digital Logger also permits pulses to be output. This means an output can be switched to On or OFF for a preset amount of time and resumes its base state after the set pulse length.

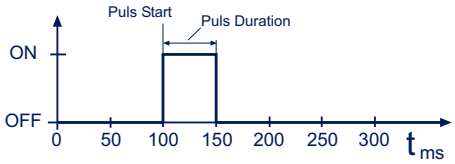
To configure the Web-IO Digital Logger for outputting pulses, select Config >> Ports >> Pulse

Required access rights: *Administrator*



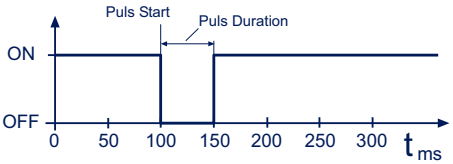
Duration
Enter here the desired pulse length in ms. A value of 1000 corresponds to a 1-second long pulse.

Pulse Polarity
positive:



In the base state the output is not switched (0V). If the output is now set to ON via browser or from an application, the Web-IO Digital Logger switches the supply voltage +Vdd to the output for the set pulse duration.

negative:



In the base state the output is switched to supply voltage +Vdd. If the output is now set to ON from an application, the Web-IO Digital Logger turns the output off (0V) for the set pusle duration.

For the Web-IO Digital Logger with relay outputs a pause of min. 200ms must be maintained between two switching operations on an output!

With negative polarity negative logic is used. Switching to ON causes the corresponding output to be turned off.

It is not possible to switch a negative pulse via HTTP from the browser. If you need to work with negative pulses from the browser, this can be done using the Java applet.

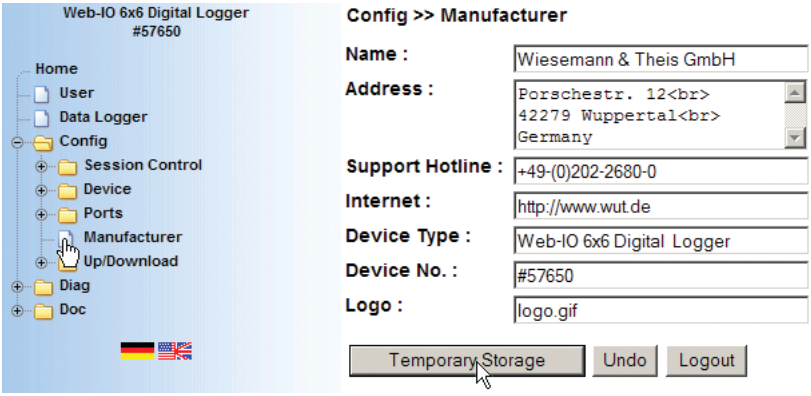
Logout and *Save* activates the new setting.

6.4 Changing device-specific displays

For project-specific OEM applications of the Web-IO Digital Logger you can change displays such as article number, manufacturer, logo, etc.

Select in the navigation tree *Config >> Manufacturer:*

Required access rights: *Administrator*



Name
Field for the company name

Address
Field for the company address

Support Hotline
Telephone number the user can call to get technical support for the Web-IO Digital Logger or for the overall project

Internet
Homepage of the company responsible for the device.

Device Type
Company-specific device name

Device No
Company-specific article number.

Logo
Path and name of the logo positioned above the navigation tree. If you specify only logo.gif, the Web-IO Digital Logger uses the internally stored logo. Alternately you can reference a JPG or GIF file located on another server.

Note that the dimensions of the logo must closely approximate those of the W&T logo (width max. 200 pixels, height max. 100 pixels) to retain clarity in the navigation frame.

Send the settings to the Web-IO Digital Logger by clicking on the *Temporary Storage* button.

Activate the new settings under *Config >> Session Control >> LogOut, Save.*

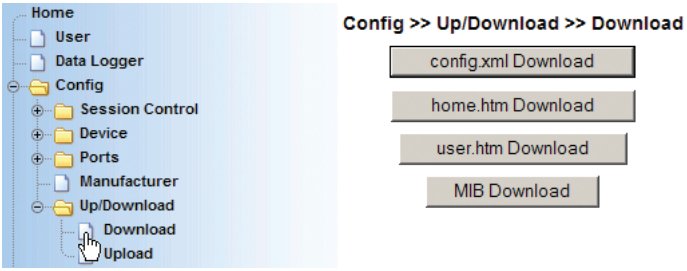
6.5 Upload/Download

The Web-IO Digital Logger allows the user to upload or download custom HTML pages, configuration data and the logo.

6.5.1 Download

To download an HTMP page or the configuration, select in the navigation tree *Config >> Up/Download >> Download:*

Required access rights: *Administrator*



Config.xml Download
The entire Web-IO Digital Logger configuration is stored in the unit as an XML file. This means that when there are projects involving multiple Web-IO Digital Loggers having the same configuration, the Administrator can make configure just one unit and load the settings into all the others.

Start the download of the XML file by clicking on the corresponding button.

The XML parameters are then displayed in the browser window. To safe the XML data as a file, most browsers allow you to right-click the source text in the text editor and from there save it.

home.html / user.html Download
Here again the selected page is displayed first in the browser window and can be saved from the source text display. For a

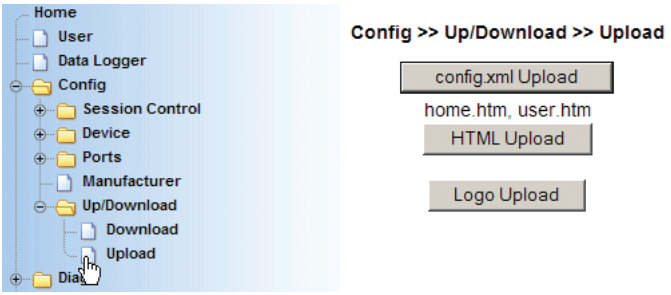
detailed description of this procedure, refer to the section *HTTP*.

6.5.2 Upload

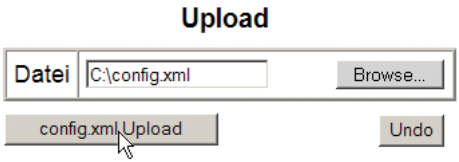
Just as you can download HTML and XML files, it is also possible to upload them to the Web-IO Digital Logger.

Select *Config >> Up/Download >> Upload:*

Required access rights: *Administrator*



config.xml Upload
The entire configuration of the Web-IO Digital Logger can be loaded into the Web-IO Digital Logger as an XML file in one operation. Clicking on the corresponding button displays an entry field in the configuration frame in which you can enter or select the upload file:



Click on the *Upload* button to start the upload.

A convenient way of saving, managing and later uploading XML files from the Web-IO Digital Logger is by using the Wutility tool.

At our Web site www.WuT.de you will find at left the Insider Search for Article No. Enter here article no. 57650, select „Tools“ in the field below, and click on „Go“. On the page which then opens select the link to the Wutility tool.

HTML Upload
The upload procedure for the home.htm and user.htm page is exactly analogous to the XML upload.

The exact procedure is described in detaili in the *HTTP* section.

logo Upload
For a custom appearance of the Web-IO Digital Logger start page, you can replace the W&T logo in the navigation frame with another one of your choice (GIF or JPG).

Clicking on the corresponding button displays an entry field in the configuration frame in which you can enter or select the upload file.

Click on the Upload button to start the upload.

For the uploads to remain in effect even after a power interruption, activate the new settings under *Config >> Session Control >> LogOut, Save.*

6.6 Diagnostics and testing

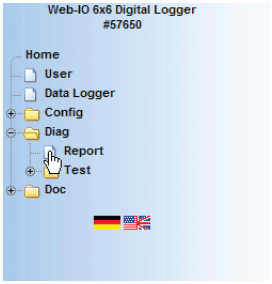
The Web-IO Digital Logger provides internal error management and a diagnostics system.

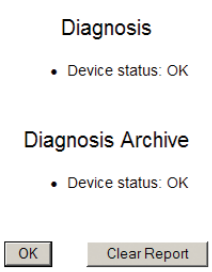
6.6.1 Error report

When an error occurs, this is indicated on the device by the *on error* LED flashing. In addition, any error states occuring are documented in a report and can be read out later.

All error messages are stored in the Web-IO Digital Logger and are retained even if the cause of the error has been remedied.

To do this, select *Diag >> Report:*





Clicking on the *Delete report* button allows the Administrator to clear all error messages.

m Required access rights: *Administrator*

A reset also deletes the report, regardless of whether it is performed by pressing the Reset button on the device, interrupting the supply voltage or by using the Reset function from the Logout page.

An exact list of possible error messages and their cause can be found in the Appendix. In the section „So-

cket programming" you are shown how errors can also be processed from your own applications.

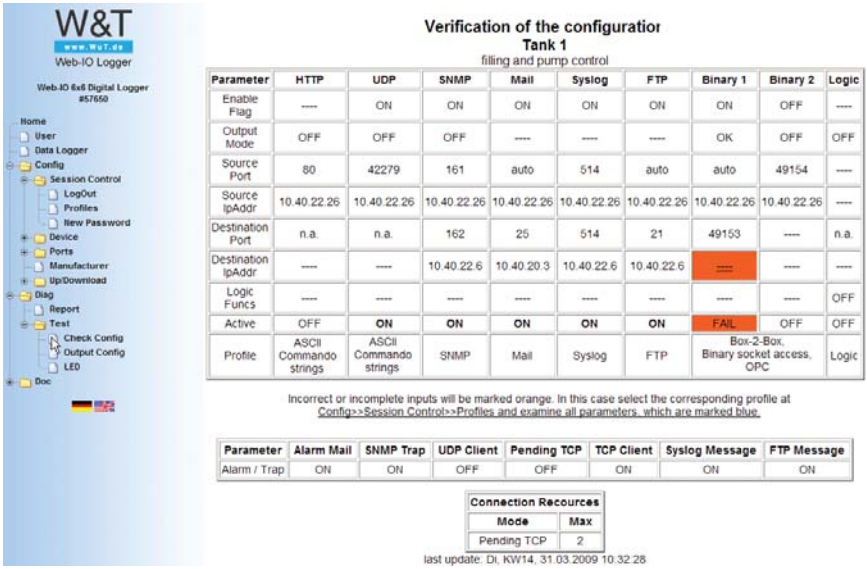
You can also process error and diagnostic messages using SNMP traps or as a Syslog system message. See the SNMP and Syslog sections.

6.6.2 Configuration

Overall view

The Web-IO Digital Logger allows the Administrator to view and check the Web-IO Digital Logger configuration on a comprehensive Web page.

Select *Diag >> Test >> Check Config*



The Web page shows which access types are activated with which parameters. Here the Web-IO Digital Logger performs a plausibility check of the settings. If missing parameters are detected which prevent proper operation of the access type, the corresponding fields are highlighted in orange.

Also checked and displayed are which transmission channels are selected for the alarms and whether all necessary parameters have been configured. Here again the access types which were not fully configured are highlighted in orange.

Finally, the number of possible static connections on Port 80 of the Web-IO Digital Logger is checked and displayed. Port 80 of the Web-IO Digital Logger can be used both for opening the Web pages as well as for socket applications.

The factory default setting is for all free ports configured so that they work as a server port on Port 80. Dynamic connections are permitted on only two ports.

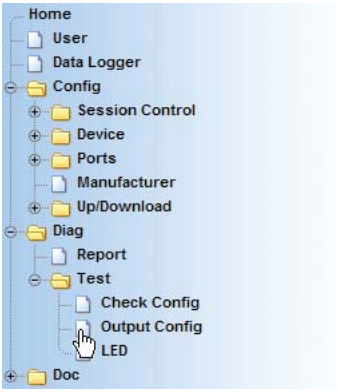
If for example 5 ports are configured on HTTP (Port 80), 3 of them can be used by continuous TCP connections, e.g., by socket applications. The remaining two free ports can accept connections, but the Web-IO Digital Logger rejects then again after a short timeout.

This ensures that you can always reach the Web-IO Digital Logger via the browser.

Overview of the output configuration
In addition to the overview of the general configuration, the Web-IO Digital Logger also offers an additional Web page which shows the configuration and status of the outputs.

Select *Diag >> Test >> Output Config*

m Required access rights: *Administrator*



Verification of the configuratior

Tank 1

filling and pump control

Output	Modus	Group Mode	Safety State
Pumpe 1	HTTP	0 1 2 3 getrennt	ON
Output 1	UDP ASCII	0 1 2 3 getrennt	OFF
Output 2	SNMP	0 1 2 3 getrennt	ON
Output 3	Binary 1	0 1 2 3 getrennt	unverändert
Output 4	Binary 2	4 5 getrennt	unverändert
Output 5	Logic	4 5 getrennt	unverändert

last update: Di, KW14, 31.03.2009 10:47:26

The following information is output for each of the outputs:

Mode
Displays the configured mode

- HTTP
- UDP ASCII
- SNMP
- Binary 1
- Binary 2
- Logic

Group Mode
If 2 or 4 outputs are combined in a group, that is indicated in this column.

Safety State
This column may contain the following entries:

- *unchanged*
  No watchdog function was set for this output.
- *ON*
  A watchdog function was set for this output.
- *Error*
  When Load Control is activated an overload or the absence of a load is detected.

In case of overload the Web-IO Digital Logger turns the output off.
(not for Web-IO Digital Logger models with relay outputs)

Since the latter case may represent a serious error, these fields are highlighted in red.

6.6.3 LED test

To test all LEDs for the input/output section of the Web-IO 12xDIgital, select *Diag >> Test >> LED* and click on the *LED Test* button.

Required access rights: *Default User*, *Config*, *Administrator*.

All LEDs are turned on for approx. 2 sec.

This function can be used to uniquely identify a Web-IO Digital Logger in the control cabinet.

The actual state of the outputs is thereby not changed. The 3 LEDs for the Ethernet connection are not affected by this test.

7 The Data Logger

- General function
- Viewing saved data
- Downloading saved data
- Clearing the Logger contents
- Formatting the Logger memory

7.1 How the Data Logger functions

The built-in Data Logger of the Web-IO Digital Logger stores up to one million events with a time stamp.

Events are considered:

- Status changes on the inputs
- Incrementing of the input counters
- Status changes on the outputs
- Alarm occurences

The minimum memory update time is 100ms. Events which occur faster than this may not be logged. For changes on the inputs the counter state of the corresponding counter is incremented. In this way you can at least determine how often the status of an input changed within the 100ms.

The built-in logger is a first-in/first-out ring memory. This means that the oldest data are overwritten when the memory is full.

The contents of the Data Logger remains stored even after power is interrupted or turned off.

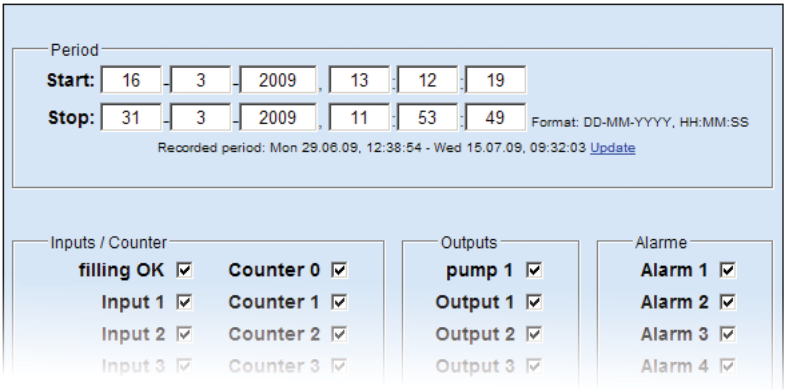
7.2 Viewing and opening the saved data

Required access rights: *Administrator*, *Operator*

To read out or display the stored data, go to the navigation tree and select *Data Logger*.

7.2.1 Specifying the display time period

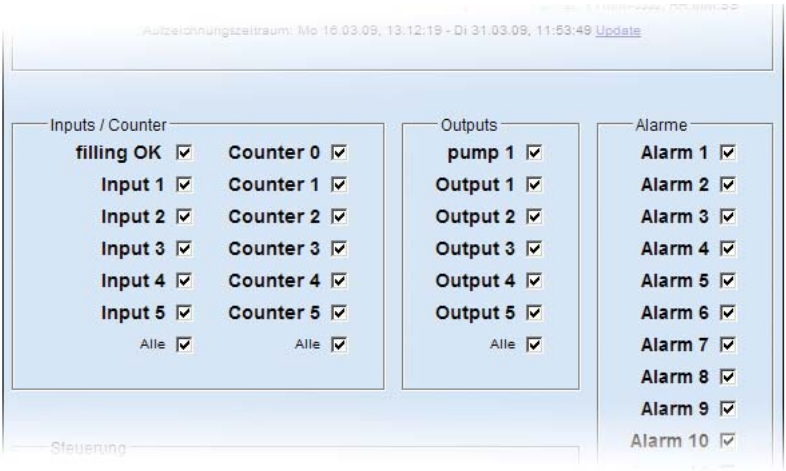
To limit the amount of data displayed, the display time period can be individually adjusted.



When opening the Data Logger page, the Web-IO Digital Logger suggests as a start time the oldest available time point and enters the time of the last saved event under *Stop*.

7.2.2 Specifying the event types to display

Using another filter criterion you can select which events to display.



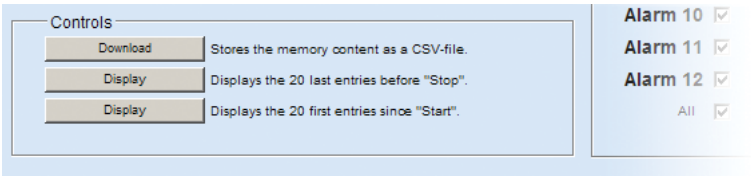
Once you have specified which inputs, counters, outputs or alarms you want to display, you can three basic ways of evaluating the Logger:

- Browser display
- Download in CSV format
- Reading from your own application

7.2.3 Displaying the Logger data in your browser

The simplest and fastest way to view the recorded data is to display them in the browser.

The user can scroll through the pre-selected time period in steps of 20 data records. You can choose whether to begin with the last 20 data records to scroll back in time from there, or display the oldest data first.



The selection is made using the corresponding button.

Date Time	Inputs						Counter						Outputs					
	I0	I1	I2	I3	I4	I5	C0	C1	C2	C3	C4	C5	O0	O1	O2	O3	O4	O5
Di 31.03.09 14:33:39.200	ON	off	off	off	off	off	17	8	6	8	6	8	off	off	off	off	off	off
Di 31.03.09 14:33:37.400	off	off	off	off	off	off	16	8	6	8	6	8	off	off	off	off	off	off
Di 31.03.09 14:33:37.100	off	off	off	off	off	ON	16	8	6	8	6	8	off	off	off	off	off	off
Di 31.03.09 14:33:36.700	off	off	off	off	ON	off	16	8	6	8	6	7	off	off	off	off	off	off
Di 31.03.09 14:33:36.400	off	off	off	ON	off	off	16	8	6	8	5	7	off	off	off	off	off	off
Di 31.03.09 14:33:36.100	off	off	ON	off	off	off	16	8	6	7	5	7	off	off	off	off	off	off
Di 31.03.09 14:33:35.800	off	ON	off	off	off	off	16	8	5	7	5	7	off	off	off	off	off	off
Di 31.03.09 14:33:35.500	ON	off	off	off	off	off	16	7	5	7	5	7	off	off	off	off	off	off
Di 31.03.09 14:33:33.600	off	off	off	off	off	off	15	7	5	7	5	7	off	off	off	off	off	off
Di 31.03.09 14:33:33.300	off	off	off	off	off	ON	15	7	5	7	5	7	off	off	off	off	off	off
Di 31.03.09 14:33:33.000	off	off	off	off	ON	off	15	7	5	7	5	6	off	off	off	off	off	off
Di 31.03.09 14:33:32.700	off	off	off	ON	off	off	15	7	5	7	4	6	off	off	off	off	off	off
Di 31.03.09 14:33:32.400	off	off	ON	off	off	off	15	7	5	6	4	6	off	off	off	off	off	off
Di 31.03.09 14:33:32.000	off	ON	off	off	off	off	15	7	4	6	4	6	off	off	off	off	off	off
Di 31.03.09 14:33:31.700	ON	off	off	off	off	off	15	6	4	6	4	6	off	off	off	off	off	off
Di 31.03.09 14:33:29.900	off	off	off	off	off	off	14	6	4	6	4	6	off	off	off	off	off	off
Di 31.03.09 14:33:29.500	off	off	off	off	off	ON	14	6	4	6	4	6	off	off	off	off	off	off
Di 31.03.09 14:33:28.200	off	off	off	off	ON	off	14	6	4	6	4	5	off	off	off	off	off	off
Di 31.03.09 14:33:28.900	off	off	off	ON	off	off	14	6	4	6	3	5	off	off	off	off	off	off
Di 31.03.09 14:33:28.600	off	off	ON	off	off	off	14	6	4	5	3	5	off	off	off	off	off	off

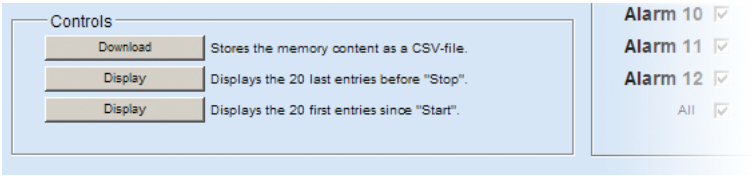
For the data shown here, display of alarms was omitted.

Inputs, counters and outputs are shown in a table. Inputs and outputs which have an ON status are highlighted in green. To scroll down in the view, you can use the buttons at the bottom of the page.

It is recommended that you reasonably limit the representation time period using the Start/Stop times to avoid unnecessary scrolling.

7.2.4 Downloading the Logger data in CSV format

To store the recorded data over a longer period of time, the Web-IO Digital Logger allows you to download them from the unit in CSV format.



The time and function restrictions are set just as for the browser display.

The CSV file can be processed in MS Excel or Open Office.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Request	16.03.2009 13:12:19.000 - 31.03.2009 15:20:00.100										
2	Report											
3	Date	Time	milli sec	filling OK	Input 1	Input 2	Input 3	Input 4	Input 5	Counter_1	Counter_2	Counte
4	16.03.2009	13:13:58	100	0	1	1	1	1	1	0	0	
5	16.03.2009	13:13:58	200	0	1	1	1	1	1	0	0	
6	16.03.2009	13:14:01	100	0	0	1	1	1	1	0	0	
7	16.03.2009	13:14:01	200	0	0	1	1	1	1	0	0	
8	16.03.2009	13:14:02	900	0	0	0	1	1	1	0	0	
9	16.03.2009	13:14:02	0	0	0	0	1	1	1	0	0	
10	16.03.2009	13:14:04	800	0	0	0	0	1	1	0	0	
11	16.03.2009	13:14:06	900	0	0	0	0	0	1	0	0	
12	16.03.2009	13:14:06	0	0	0	0	0	0	1	0	0	
13	16.03.2009	13:14:07	700	0	0	0	0	0	0	0	0	
14	16.03.2009	13:14:08	800	0	0	0	0	0	0	0	0	
15	16.03.2009	13:36:54	900	1	0	0	0	0	0	1	0	
16	16.03.2009	13:36:54	0	1	0	0	0	0	0	1	0	
17	16.03.2009	13:36:54	200	0	0	0	0	0	0	1	0	
18	16.03.2009	13:36:55	300	0	0	0	0	0	0	1	0	
19	16.03.2009	13:36:55	500	1	0	0	0	0	0	2	0	
20	16.03.2009	13:36:55	700	0	0	0	0	0	0	2	0	
21	16.03.2009	13:36:55	900	1	0	0	0	0	0	3	0	

Note when restricting the amount of data that Excel can handle a maximum of 65536 lines depending on the version

7.3 Reading Logger data from your own application

By means of a socket connection you can also read out the contents of the Logger from your own applications.

This is done using one of the two binary accesses with special data structures.

A detailed description of this procedure is described in the section on socket programming beginning on page 155.

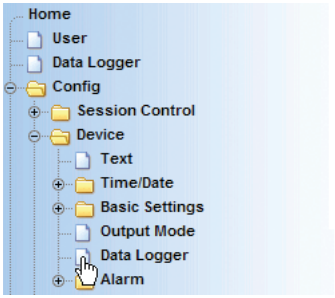
7.4 Erasing and formatting the Data Logger

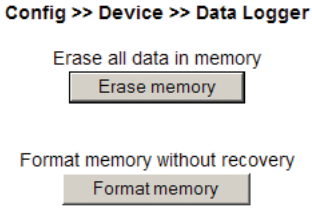
As already mentioned, the Data Logger is a ring buffer whereby the oldest data are overwritten when it becomes full.

Depending on the application it may also make sense to specifically clear the entire memory before starting a new project for example.

Required access rights: *Administrator*

From the navigation tree select *Config >> Device >> Data Logger*.





Use the *Erase memory* button to completely clear the Data Logger.

This resets only the read and write pointers of the Logger.

If you accidentally erase the memory, it may be possible to restore the deleted data by returning the unit to W&T.

Use the *Format memory* button to reset the complete data range to 0. In this case the data can no longer be restored.

8 Appendix

- Firmware update
- Emergency access
- Port numbers used, network security
- Error messages and their causes
- Technical data
- Declaration of Conformity
- Navigation tree reference
- Index

8.1 Firmware Update

The operating software of the Web-IO Digital Logger is under constant development. The following section describes how to perform an upload for the latest firmware version.

8.1.1 Where can I get the current firmware?

On our web site www.WuT.de, having selected the Englisch language version, on the left side you find „Insider search via article number". Enter here the article no. 57630, select „Firmware" in the checkbox below and click „Go". On the page that will open, you select the link to the latest firmware.

8.1.2 Firmware update over the network using WIndows

For this you will need a PC running under Windows 9x/NT/2000/XP with a network connection and activated TCP/IP stack. For the update process you will need two files which, as already described, are available from the Homepage at *http://www.wut.de*.

- The executable *WuTility* -Tool for sending the firmware to the Web-IO Digital Logger
- The file with the new firmware you want to send to the Web-IO Digital Logger

No special preparation of the Web-IO Digital Logger is necessary for performing the firmware update.

The *WuTility* used for the update recognizes all W&T devices located in your network and is for the most part self-explanatory. If you do have questions or anything is unclear, please refer to the associated documentation or use our online help.

Be sure to use only the current version of Wutility for the update. Use of older Wutility versions may render the Web-IO Digital Logger inoperable.

We recommend deleting all passwords and performing a Reset of the Web-IO Digital Logger before the update is loaded.

For the update Wutility requires unrestricted network access to the Web-IO Digital Logger. Software firewalls and Internet security tools should be configured so that access to the Web-IO Digital Logger is not prevented.

Never intentionally interrupt the update process by disconnecting power or actuating the Reset button. After an incomplete update the Web-IO Digital Logger will be non-operational.

Never mix files with different version numbers in the file name. This will render the device inoperable.

The Web-IO Digital Logger normally recognizes when transmission of the new operating software is complete and then automatically performs a reset. In rare cases it may be necessary after an update to restart the Web-IO Digital Logger using a hardware reset (Reset button of Power ON). A software reset via network access through Port 8888 will not then be sufficient.

8.2 Emergency access

The serial port (DTE) of the device provides an emergency access. Activiate this as follows:

Use a serial cable to connect the device to a PC (null modem connection) and start a serial terminal program. Use the following program settings:

9600,8,N,1,no handshake

Press the Reset button on the device. As soon as you hear the tone, press the following letters on your keyboard 3 times for the individual accesses:

3x „u“ Opens the update port. You can now load a firmware update

3x „f“ Resets the device to its factory default settings.
All previous configuration settings (including the IP address) are lost.

3x „p“ Deletes all previously assigned passwords.

For confirmation the **system error** and the **on error http** LEDs flash several times in succession.

3x „x“ (Enter directly after pressing the Reset button, before the audible tone) assigning/changing the IP address. At the prompt, enter the desired IP address.

8.3 Ports and network security

8.3.1 Ports

Following is an overview of all the ports used in the Web-IO Digital Logger Digital Logger.

Port number	Protocol	Application	changeable	suppressable
20	TCP	FTP-Data	1...65535	yes
21	TCP	FTP-Control	1...65535	yes
23	TCP	Telnet [2]	no	no
25	TCP	SMTP Mail [1]	no	yes
37	TCP	Query to Time-Server [1]	no	yes
53	UDP	Query to DNS-Server [1]	no	yes
69	TCP	Update via TFTP [3]	no	yes
80	TCP	List port TCP/http	yes	no
110	TCP	SMTP after POP3 [1]	no	yes
161	UDP	SNMP [3]	no	yes
1111	TCP	Configuration of serial port [2] [3]	no	no
8000	TCP	Socket access for serial Port [2]	yes	no
8009	TCP	Auxiliary port for manual	no	yes
8512	UDP	Inventorying [1]	no	no
8513	UDP	Inventorying [1]	no	no
8888	TCP	Reset [3]	no	no
9084	TCP	Port-Reset [2] [3]		
9094	TCP	Box-Control-Port [2] [3]	no	no
42279	UDP	UDP Socket	1...65535	yes
45889	UDP	Binary 1	1...65535	yes
45889	UDP	Binary 2	1...65535	yes
49153	TCP	Binary 1	1...65535	yes
49154	TCP	Binary 2	1...65535	yes

[1] No write access to the Web-IO possible
[2] only Web-IO 57631 with Com-Server function
[3] password protected

Each port number may be used for only one service in the Web-IO Digital Logger. If differing numbers are used with the changeable ports, be sure that no port number is used twice.

8.3.2 The W&T Web-IO Digital Logger and network security

Security in networks is justifiably gaining increased attention. All the experts agree that there can be no such thing as absolute security at today's state of technology. Each customer must therefore find an appropriate balance between security, functionality and cost for his specific requirements.

To give the customer the greatest possible degree of flexibility in this regard while responding to changing security requirements from a pure testing and installation environment to critical production applications, the security measures offer a high degree of user configurability.

This document provides an overview of the security measures implemented on the Wiesemann & Theis Web-IO Digital Logger and how they can be used. It is assumed that the original W&T firmware (without custom modifications) is used. For additional details, refer to the respective sections in this Manual.

8.3.3 The authorization concept of the Web-IO Logger:

As already described in the section Web-Based Management, the Web-IO Digital Logger uses three authorization levels:

- Default User (with no special login)
- Operator
- Admin

These users normally access Port 80 through the Web interface, as is customary with http.

The default user can use the Web interface to read Web-IO Digital Logger data, and has access to the diagnostic tools as well as some online documentation (abbreviated manual, data sheet for the device, support information).

The operator can also set the outputs and configure the alarm outputs. No other settings can be changed by the operator.

The Administrator has full access to all the functionalities of the device. He specifies the operation mode(s) used and the network parameters.

Passwords can be defined for the *Operator* and *Admin.* The device does not require that passwords be used, nor that they must be different for the *Operator* and *Admin*, nor the minimum length, composition or regular changing of othe passwords. This is left to the user as he sees fit.

No encoding is used to send the password to the Web-IO Digital Logger. You should therefore ensure that password-protected access is only over an Intranet which the customer considers to be secure. For accesses over the public Internet, additional measures such as a VPN tunner (Virtual Private Network) may need to be taken. But this is a general problem of network security, and every customer needs to find his own solutions.

8.3.4 Ports with special function:

In addition to access over the Web interface, a variety of functions can be activated through various TCP or UDP ports. These are shown in Table 1 (Excel table). A series of functions can be turned on and off, but only with „Admin“ rights. The basic recommendation is to turn off all the functions you are not using. More on this subject can be found in the Manual.

SNMP:
To be able to incorporate the Web-IO Digital Logger into an SNMP-based systems management, all the essential configuration settings can also be modified via SNMP. Access is protected by using the Admin password in SNMP as a *community string*. SNMP access can be turned off.

Ports for inventorizing using the „WuTility“ tool:
Like all the „intelligent“ components from W&T, the Web-IO Digital Logger can be accessed using the „WuTility“ tool. The tool can use ports 8512 and 8513 to read information such as hard- and firmware version, IP address, etc. This access cannot be

turned off, but write access to the device is not possible with the tool.

Firmware-Update:
The TFTP (TCP port 69) protocol makes it possible to get a firmware update and overwrite all the configuration settings. Only, however, if update mode was previously enabled. There are several ways to do this:

- One-time activation using a special http string in connection with the Admin password (if set)
- One-time activation using *Config >> Session Control >> Log-out* and the *Firmware Update* button

Reset using Port 8888:
Opening a connection (e.g., using Telnet) to Port 8888 allows the device to be reset. Here you must give the Admin password (if set) right after the connection is opened. This immediate sending of the password is impractical by manual means, and should be performed by a program. The consequences of a reset are the same as for a momentary interruption of power or pressing the Reset button on the device.

8.3.5 Restoring the factory defaults

Restoring the factory faults allows you to undo all the security settings made previously. Among other things the Admin user is no longer password protected. This option thus represents also an „emergency access“ for cases where the password has been forgotten or in emergencies where the persons who know the passwords are not available.

Admin authorization is required to reset the factory defaults via software over the network. The factory defaults can however be restored through the serial port of the Web-IO Digital Logger by pressing a key combination after powering up or after a reset.

In any case you should be sure that only authorized persons have physical access to the unit including the serial port, which may mean keeping the device in a lockable control cabinet.

8.4 Network inventory via UDP

Larger installations especially often require inventorizing of the Web-IO Digital Loggers or also W&T Com-Servers currently located in the network. UDP Port 8513 is provided for this purpose. After receiving a network packete on this port, the Web-IO Digital Logger replies to the sender with an info packet. This contains information about the Web-IO Digital Logger and its network parameters.

Whether a packet was received via broadcast or directly addressed makes no difference in generating the info structure. The only requirement is that Port 8513 be used.

For reasons of backward compatibility the Web-IO Digital Loggers continue to support UDP info port 8512. For new applications, however, only Port 8513 should be used.

8.4.1 The info packet

Each info packet consists of 3-6 data structures: *BOX_VERSION* contains information about the respective model of the Web-IO Digital Logger or Com-Server and its firmware version. Network-relevant data such as MAC and IP address can be taken from the *BOX_DESCRIPT* structure. The 3.-6. structure *PORT_DESCRIPT* provides information about the sete operation mode (for W&T Com-Servers only) and the current connection status of the individual ports. The total length of an info packet is calculated as follows: *10 + 22 + (port_anz × 10) bytes*

```
#pragma pack(1)

typedef struct _BOX_VERSION             // (10byte)
{       unsigned int version;           // 0x10: 1.0 (Version of this structure)
        unsigned int sw_rev;            // z.B. 1.24 (0x1501)
        unsigned int hw_rev;            // C2_EURO          = 2.0 (0x0002)
                                        // C3_PC104         = 3.0 (0x0003)
                                        // C4_MINI          = 4.0 (0x0004)
                                        // C5_100BT         = 5.0 (0x0005)
                                        // C6_INDUSTRY      = 6.0 (0x0006)
                                        // C8_LOW_VOLTAGE   = 8.0 (0x0008)
                                        // C8_WEB-IO            = 8.1 (0x0108)
        unsigned int reserved[2];
```

```
 }BOX_VERSION;

typedef struct _BOX_DESCRIPT            // ( 22 byte)
{         unsigned char mac_addr[6];    // MAC address of the Com-Server
          unsigned long IP_number;      // IP address of Com-Server
          unsigned long gateway;        // Gateway
          unsigned long subnet_mask;    // Subnet Mask
          unsigned int MTU;             // MTU
          unsigned int port_anz;        // Number of ports in the Com-Server
 }BOX_DESCRIPT;                         // (port_anz * 10 byte)

typedef struct _PORT_DESCRIPT           // (10 bytes)
 {union
          { unsigned int wState;
            struct _new_type
             { unsigned char port_type;   // 0x01 = serial port (Serial CPU-Port)
                                          // 0x02 = serial port (UART Port)
                                          // 0x03 = reserved
                                          // 0x04 = Digital I/O
                                          // 0x05 = Analog I/O (in development)
                                          // 0x06 = W&T Dual Port RAM
               unsigned char state;       // 0=free, 1=connect, 2=waits
             };
          };
          unsigned int mode;            // 0x0001 = TCP-Client
                                        // 0x0002 = TELNET-Client
                                        // 0x0003 = FTP-Client
                                        // 0x0004 = Box2Box-Client (aktiv)
                                        // 0x0005 = UDP Send/ReceiveMode
                                        // 0x0006 = MULTI PORT (DPRAM, SERIAL PROTOCOL)
                                        // 0x0007 = SNMP-Agent
                                        // 0x0008 = Box2Box Server (passive)
                                        // 0x0010 = SLIP Router
                                        // 0x0020 = PPP Router (in development)
                                        // 0x0030 = Box2Box Bus System: Slave Box
                                        // 0x0040 = Box2Box Bus System: Master Box
          unsigned long remote_IP;      // if state == CONNECT, otherwise 0
          unsigned int remote_port;     // if state == CONNECT, otherwise 0
 }PORT_DESCRIPT;

typedef struct _WT_INTERN3              // ** all parameters in Hostorder / Low Byte First **
{         BOX_VERSION   bv;             // Port = UDP_BOX_INFO_8513
          BOX_DESCRIPT  bd;
          PORT_DESCRIPT pd[ACT_PROCESS];
 } WT_INTERN3;

#pragma pack()
```

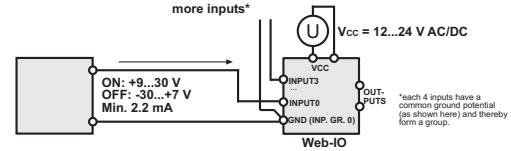
The structure PORT_DESCRIPT does not have to be evaluated for the Web-IO Digital Logger. It is intended only for information pertaining to W&T Com-Servers.

All variables of type integer *and* long *are represented in host order. This means the low bytes come first. The IP-Adresse 172.17.2.3 appears for example in byte sequence 3, 2, 17, 172.*

UDP packets are connectionless and unsecured datagrams. Especially when using with broadcasts, both your own request packet as well as the reply packet from the Com-Server can become lost. To be sure you have gotten all the Com-Servers installed in a subnet, the request broadcast should therefore be repeated as necessary.

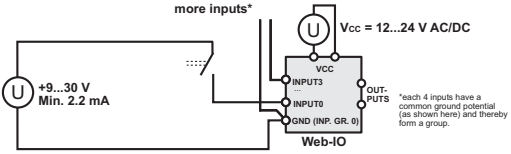
8.5 Wiring examples for the inputs

Connecting a current sourcing device



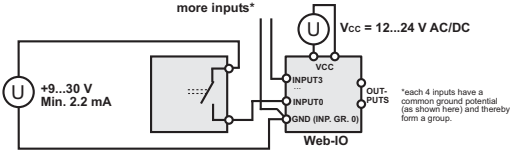
Attachment circuit for Web-IO inputs: Connecting a device with current sourcing output
(Example: Input 0 belongs to input group 0)

Connecting a switch



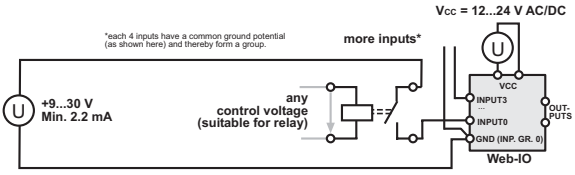
Attachment circuit for Web-IO inputs: Connecting a switch
(Example: Input 0 belongs to input group 0)

Driving with a potential-free contact



Attachment circuit for Web-IO inputs: Connecting a device with a potential-free contact
(Example: Input 0 belongs to input group 0)

Driving with a series connected relay



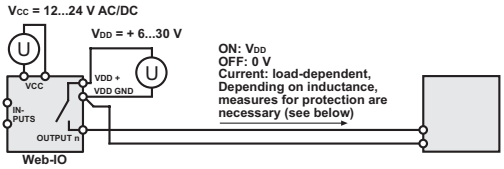
Attachment circuit for Web-IO inputs: Connecting any kind of signal by a relay
(Example: Input 0 belongs to input group 0)

**Each 4 inputs have (as shown here) a common ground reference and therfore form a group*

8.6 Wiring examples for the outputs

Switching loads up to 500mA

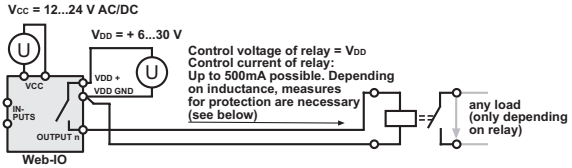
(not applicable to Web-IO with relay outputs)



Attachment circuit for Web-IO outputs: Connecting a device with current consumption up to 500mA

The outputs which are grouped together for increased switching load capacity must be connected together a s agroup in the output configuration.

Switching loads through a relay



Attachment circuit for Web-IO outputs with relays for potential-free contact

A suitable method for an inductive load could be for example a freewheeling diode. Thie diode should be dimesioned to take over at least 100 times the current usually flowing through the relais coil.

8.7 Error messages and causes

The following list provides an overview of all the error messages that can be shown in the report under *Diag >> Report*.

Error message	Possible cause
Format error in the DNS request	DNS server does not support the request format used by the Web-IO
Internal DNS server error	Unknown error in name resolution
DNS: Name does not exist	No IP address could be determined for the host name requested from the DNS server, for example the time server.
DNS request type not supported	DNS server does not support the request format used by the Web-IO
DNS request was rejected	The DNS server rejected the request. The reason may be an overload of the DNS server.
DNS server cannot be reached	No contact could be established with the DNS server, for example because there is a problem with the network infrastructure
DNS server not recognized	Reply from DNS server not received or format not recognized
DNS: Incorrect IP address	No DNS server can be reached under this IP address
DNS: No resolution possible	Requested name not known
Time Server: Synchronization failed	No time information received from the time server
Mail: Server denying access	Login to SMTP server was rejected (no access rights?)
Mail: Sender could not be transmitted	Format of the sender address not accepted
Mail: Recipient could not be transmitted	Format of the recipient address not accepted
Mail: Switch to data mode failed	Mail server could not correctly process e-mail
Mail: Mail text could not be transmitted	Mail server could not correctly process e-mail text
Mail: Mail server connection failed to close	Mail server did not acknowledge closing of connection
Mail server not responding	No connection to mail server possible (missing DNS entry, network down, …)
Mail: IP address or name of the mail server not entered	see Config >> Device >> Mail Admin
Sending of the mail was interrupted and is being repeated	Sending of the e-mail will be automatically repeated after one minute
Transmission error: Error code xxxxx	Internal communications errror. If the error occurs again after Power-On-Reset, contact your dealer and provide the error code.
Protocol error: Error code xxxxx	Internal communications errror. If the error occurs again after Power-On-Reset, contact your dealer and provide the error code.
No router was configured	No Gateway entry (Config >> Device >> Network)
Destination IP address unknown: xxx.xxx.xxx.xxx	Communications partner cannot be reached under indicated IP address
A network fault was detected (cable open or no link)	Cable open or no link (no connection between Web-IO and HUB)
The COM Server triggered a reset	Reset due to external fault
Warning: Watchdog Timer Reset triggered	Reset due to internal fault

Error message	Possible cause
UDP port number invalid: use Port 42279	Duplicate assigned port number, port number greater than 65535 or text given instead of digits for port number
Checksum error: factory defaults were loaded	ROM error detected in configuration area
SNMP: IP address or name of Snmp manager not entered	see Config >> Device >> Alarm >> Alarmx >> SNMP
UDP alarm: IP address or name of UDP server not entered	see Config >> Device >> Alarm >> Alarmx >> UDP
UDP alarm: Port of UDP server not entered	see Config >> Device >> Alarm >> Alarmx >> UDP
Number of pulses exceeds 2000/sec: pulse mode input x stopped!	Input signal frequency too high
50xxx-Mode: Invalid UPD IP address	see Config >> Device >> Mode 50xxx >> UDP Client
50xxx-Mode: Invalid TCP IP address	see Config >> Device >> Mode 50xxx >> TCP Client
Watchdog timer expired	Safety watchdog detected no network traffic within the specified timeout time and the outputs were set to Safety State
Box2Box: Invalid port number	Port number greater than 65535 or text entered instead of digits for port number. Config >> Device >> Box 2 Box
Box2Box: Invalid slave IP address	No IP address specified or invalid address format. Config >> Device >> Box 2 Box
Port number assigned more than once	Each mode must use a unique port number (see Appendix 7 Port Number List)
50xxx: Invalid local port number	Duplicate assigned port number, port number greater than 65535 or text entered instead of digits for port number. Config >> Device >> Mode 50xxx
Box2Box: Invalid local port number	Duplicate assigned port number, port number greater than 65535 or text entered instead of digits for port number. Config >> Device >> Mode Box2Box
Box2Box: Slave access denied, please check Box2Box settings	Slave many be in Operator or Administrator Login, slave working in 50xxx mode or is blocked by another mode.
Box2Box: Slave is in Box2Box Master Mode	see Config >> Device >> Box 2 Box for Slave Web-IO
Box2Box: Master could not be cleared, please log out	Slave is in Operator or Administrator Login and could not clear the master connection data
Overload: Details under Diag>>Test>OutputConfig	An overload or no load was detected on one or more outputs
Restart using Delete report	After a load error the corresponding output is enabled again using Delete report. After an error caused by high-frequency input signals, pulse mode is activated again after Delete report.

8.8 Technical Data

8.8.1 Technical Data 57650

Network:	Ethernet 10/100BaseT autosensing
Protocol:	TCP and UDP Sockets, Client and Server SNMP incl. Traps SMTP (E-Mail), FTP, Syslog OPC Server Inventory, group management
Latency times:	The Web-IO 12xDigital is designed for signal change times > 50ms. The time behavior is also heavily dependent on the network load and network infrastructure.
Digital outputs:	12 x Digital Out 6V-30V, 500mA Grouping à 2 or 4 outputs Max. group current 2A Max. total current 6A Minimum load 22kOhm Thermal short circuit protected
Digital inputs:	12 x Digital In, Max. input voltage +/-30V Reverse polarity protected within this range Switching threshold 8V +/- 1V "On" current = 2.2 mA
Terminations:	2 x 16 screw terminals
Galvanic isolation:	Digital outputs - Network: min. 500 V to network Digital inputs: min. 1000 V
Serial port:	Configuration port with RS232 interface DB9-plug, configured like PC-COM
Baud rate:	9600 baud
Data format:	8 data bits, 2 stop bits No Parity
Flow control:	No Handshake
Displays:	Status LEDs for network 24 LEDs for digital states
Operating elements:	1 hidden reset button
Supply voltage:	12-24V AC/DC
Storage temperature:	-25°C - 70°C
Operating temperature:	0°C - 60°C
Housing:	Plastic housing for top-hat rail mount 106.8 x 87.8 x 62.6 (l x w x h)
Weight:	approx. 260 g

Index

Symbole

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

Y

Z