# W&T

# Manual

Installation, Startup and Applications

## Web-IO Analog 4.0

valid for Web-IO models:
#57761    Web-IO 4.0 Analog 0-20mA
#57762    Web-IO 4.0 Analog 0-10V

## Content

# 1. Quick Startup

## Network connection



## Power supply



24V DC
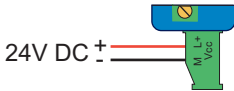
or PoE

## IP addressing

Install WuTility (Download: http://wut.de/wutility).

After starting WuTility your Web-IO will appear in the list. If more than one device is shown, identify your device by its Mac address

(white sticker "EN = 00c0:3d......"")

If there is a DHCP server is your network, you can use the assigned IP address for an initial test. Using you IP address icon in WuTility you can instead assign an available static IP address.

## Function test

In your browser open the Web page of the Web-IO using the address *http://<IP Address des Web-IO>*.

# 2. Product Introduction

## Hardware equipment

The Web-IO devices differ depending on the model in their hardware equipment:

### #57761 - Web-IO 4.0 Analog 0-20mA

Network interface:          RJ45 10/100BaseT,
                            Power over Ethernet - PoE
Power supply:               Screw terminal 12 .. 48V or PoE
IO channels:                Two channels selectable as input:
                            0/4-20mA passive /active (active
                            voltage Umax 14V / 40mA)
                            or output:
                            0/4-20mA, burden max. 500Ohm,
                            supply 10V guaranteed)

### #57762 - Web-IO 4.0 Analog 0-10V

Network interface:          RJ45 10/100BaseT,
                            Power over Ethernet - PoE
Power supply:               Screw terminal 12 .. 48V or PoE
IO channels:                Two channels selectable as input:
                            0-10V, 1MOhm
                            or output: 0-10V, Imax 20mA

## Network security

The Web-IO has an internal firewall. All available network access points can be configured and must be first enabled by the Administrator.   By default only browser access, inventorying via Wutility, and the port for initializing firmware updates are enabled. DHCP is also enabled.

For all communication paths you can explicitly specify whether the outputs may be accessed.

A list of the currently open TCP and UDP ports can be found in the navigation tree under *Port-List*.

## Access rights

The Web-IO is configured and operated from the browser. There are three authorization levels for access:

### Guest
The guest can read the status of inputs and outputs without logging in.

### User
The user can switch the outputs after logging in with a password if they are enabled for browser access.

### Administrator
The administrator has unlimited configuration and access rights after logging in with a password.

*By default no passwords are assigned. Simply click on the Login button.*

After logging in, the enabled configuration areas can be opened on the left side of the navigation tree. Help and information about the respective configuration options can be obtained using the Info buttons on the right side.

Clicking on the Apply button makes the settings immediately

active.

*All other descriptions affecting the configuration require Administrator access.*

# Application and access possibilities

### Browser access
On the homepage you can monitor the states of inputs and outputs. The outputs can also be set by users with the appropriate access rights.

In addition a fully customized Web page can be uploaded to the device and saved.

### E-mail sending
The Web-IO allows you to send e-mail messages depending on IO states or at fixed intervals. The Web-IO also supports authentication procedures prescribed by public providers.

### Box-to-Box
Two Web-IOs can be configured so that the outputs from the first Web-IO follow the inputs of the second. This work in both directions when configured as such.

### Integration into existing systems
To integrate into existing systems the Web-IO supports (when configured as such) communication via several selected standard protocols.

## MQTT

In terms of Industry 4.0 and the Internet of Things MQTT is an innovative communication method. The Web-IO can use an MQTT Publish to inform an MQTT Broker of the status of the IOs and use MQTT Subscribe to accept the request to set the outputs.

## REST

REST (Representational State Transfer) is another Web-based protocol which can be used to integrate the Web-IOO into the environment of Industry 4.0 and the IoT.

## Web-API - HTTP-Requests/AJAX

The status of inputs and outputs can be queried using HTTP requests. In addition the outputs can be directly controlled using HTTP requests.

## OPC

In combination with the W&T OPC Server the Web-IO can be accessed from any OPC client application.

## SNMP

Both the state of the inputs and outputs as well as the configuration and error status can be called via SNMP. A private MIB for simple incorporation into SNMP systems is available for direct downloading from the device.

*http://<ip-adresse>/mib.zip*

## Modbus-TCP

With Modbus-TCP the Web-IO supports one of the most common industry protocols. Any Modbus-TCP master can access the IOs by reading and writing the corresponding registers.

**Custom applications**

The Web-IO offers TCP and UDP socket accesses for accessing from your own applications. In both cases the Web-IO supports access using readable command strings or exchanging binary structures.

Since HTTP requests are also supported, you can also access the Web-IO from your own applications (e.g. with PHP or JavaScript).

## Alarms/messages

Depending on predefined events on the IOs the Web-IO can issue messages, such as sending an e-mail. Other actions include sending syslog messages, SNMP traps or MQTT publishes, writing to a file via FTP, sending of data via TCP or UDP, and setting your own outputs.

# 3. Installation and Wiring

Installation and wiring of the Web-IO requires qualified techni-cians. The generally applicable rules of technology and the respective regulations and standards must be observed.

## #57761 - Web-IO 4.0 Analog 0-20mA

### Installation
The Web-IO 4.0 Analog 0-20mA is intended for installation in the control cabinet. The Web-IO should be snapped on to a 35mm DIN rail per EINI EN 50022. The Web-IO takes up 22mm of width.

### Wiring

Ethernet
connector (PoE)

Device status LEDs

In-/Output wiring

**12**
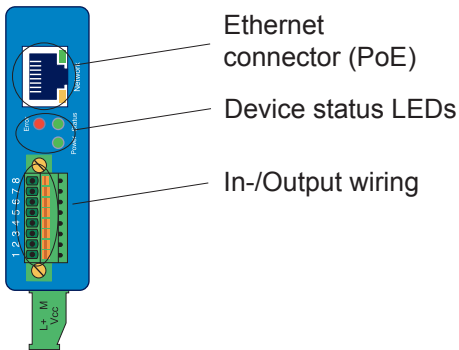
## Connecting the supply voltage

The Web-IO is powered either by PoE (Power over Ethernet4 Class 2) or with a DC voltage between 12 and 48V. The supply voltage is connected using the green terminal on the underside.

*When externally powering the Web-IO Analog only potential-free power supplies may be used. Their reference ground for the output voltage must not have a direct connection to the PE.*

*Simultaneous connection of an external power supply and a PoE infrastructure is not permitted.*
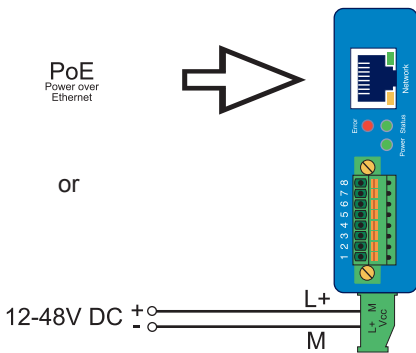


At a typical supply voltage of 24V the Web-IO draws approx. 100mA of current.

## Wiring the in-/outputs

Depending on the configuration the Web-IO Analog can be wired as follows:

## Terminal assignments for model 57661

| Terminal | Function |
|----------|----------|
| 1 | Current output |
| 2 | Current input |
| 3 | GND |
| 4 | --------------------------- |
| 5 | Current output |
| 6 | Current input |
| 7 | GND |
| 8 | --------------------------- |

## Current input 0/4-20mA, passive
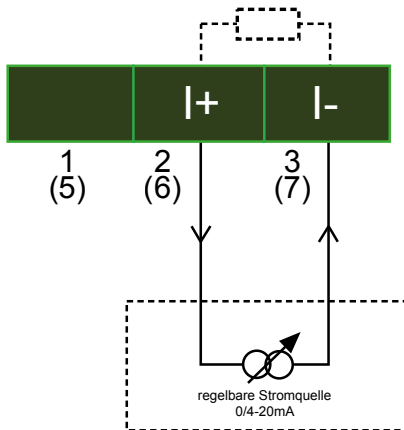
## Current input 0/4-20mA, active

For active current input 14V is provided on Pin 1 and 5.

Umax 14V

| I-Out | I-In | GND |
|-------|------|-----|
| 1 (5) | 2 (6) | 3 (7) |

max. load 500Ω

## Current output 0-20mA

Umax 10V

| I+ | I- | GND |
|----|----|-----|
| 1 (5) | 2 (6) | 3 (7) |

max. load 500Ω
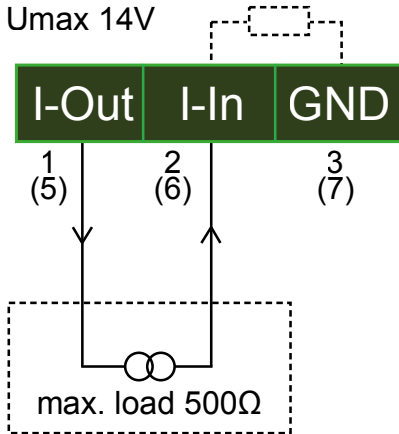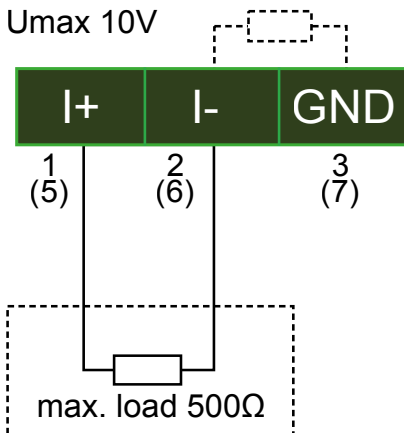
## Terminal assignment for model 57662

| Terminal | Function |
|----------|----------|
| 1 | Voltage output |
| 2 | Voltage input |
| 3 | GND |
| 4 | --------------------------- |
| 5 | Voltage output |
| 6 | Voltage input |
| 7 | GND |
| 8 | --------------------------- |

## Voltage input, passive 0-10V
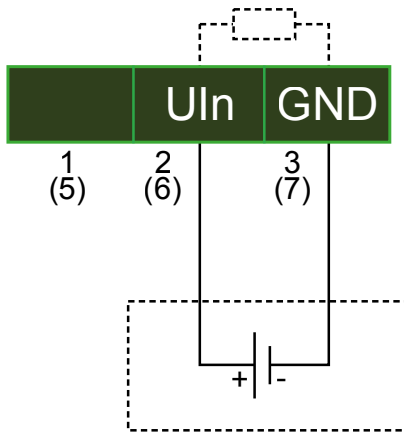
## Voltage input, active 0-10V

For active voltage input 14V is provided on Pin 1 and 5.

Umax 14V

| UIn | GND |
|-----|-----|

| 1 (5) | 2 (6) | 3 (7) |
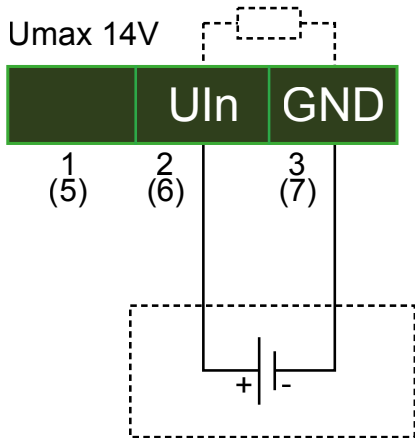
## Voltage output 0-10V

The device requires a Sense line to check the output signal and thereby regulate the voltage output.

For this the device has an internal Sense line which is used by default.

Imax 20mA

| UOut | | GND |
|------|--|-----|

| 1 (5) | 3 (7) |
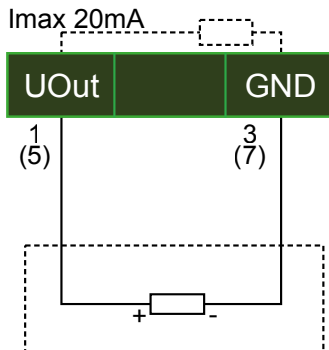
You may also connect this Sense line externally. This is recommended especially for longer cable routing.

Imax 20mA

| UOut | Sense | GND |
|------|-------|-----|
| 1 (5) | 2 (6) | 3 (7) |
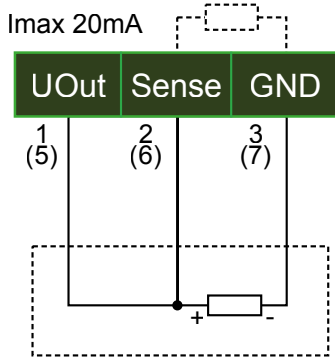
## Network connection

A conventional Ethernet patch cable (min. CAT5) with RJ45 plugs can be used for the network connection.

V-
V-
Rx-/V-
V+
V+
Rx+/V-
Tx-/V+
Tx+/V+

RJ45

Link

Speed

■ Data
■ Spare pair supply
■ Phantom supply
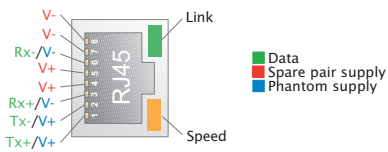
For PoE (Power over Ethernet) capable infrastructures the Web-IO can be powered through the network connection.

# 4. LED Indicators
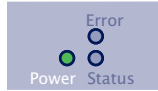


## Power-LED

OFF: There is no power. Check for proper connection of the supply voltage, either PoE or the external power supply.

ON: There is power from PoE or the external power supply.
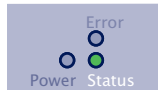


## Status LED

Rapid flashing = Booting up, no IP

Rapid flashing (approx. 3x/s) indicates that the device is booting up and/or has not been assigned an IP address. Please use WuTility for example to assign the device an IP address.

Slow continuous flashing =  Ready

Slow continuous flashing (approx. 1x/s) indicates that the device is ready. After configuring the IP address the start page of the device can be opened after entering the IP address in a web browser.

Subject to error and alteration
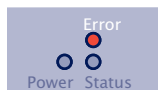
**Error LED**
The Error LED indicates error conditions on the device.

All LEDs on = Self-test error

The self-test performed after each startup or reset could not be correctly finished. In this state the device is no longer functional.

This error can occur when a software update was prematurely stopped and not all the operating software was transmitted. Repeat the software update over the network (see Firmware Update section) and address the device with the assigned IP address.

If the error cannot be re solved, there may be a hardware problem.



Speed (yellow)

OFF: Simultaneous on/flashing of the Link/Activity LED means there is a link to a device at 10MBit/s (10BaseT).

ON: Simultaneous on/flashing of the Link LED means there is a link to a device at 100MBit/s (100BaseT).



Link/Activity (green)

OFF: The device is not detecting a Link pulse from a hub/switch. Check the cable or the hub port.

ON: The device has a valid link to a hub/switch. In this case the

Speed LED indicates the speed.

Flashing: The device is receiving/sending network packets

Additional LEDs (internal)
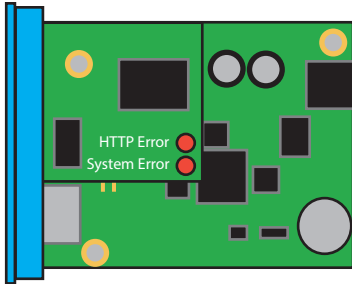


HTTP Error: Indicates internal errors in the configuration. For troubleshooting open the page http://xxx.xxx.xxx.xxx/diag in the device.

System Error: Serious hardware fault. Try to restart the device by cycling power. If the condition remains unchanged, please return the unit for analysis

⚠️ *If the Web Thermometer has no IP address or address 0.0.0.0, the "HTTP Error" and "System Error" LEDs remain on after a reset or restart! Only if an IP address is assigned do the LEDs go off.*

# 5. Startup

Once the Web-IO has been properly installed and wired the supply voltage can be turned on. All three status LEDs should come on briefly. After approx. 5 seconds only the Power LED and the Status LED should flash at 1 sec. intervals. If there is a functioning network connection the green LED in the network socket indicates an existing link. The orange LED indicates the network speed.

On = 100MBit/s

Off = 10MBit/s

## Assigning the IP address

As shipped the Web-IO has IP address 0.0.0.0 and DHCP is enabled.

### Networks with DHCP
If a DHCP server is active in the same network in which the Web-IO is connected, an IP address should be automatically assigned to the Web-IO. To be able to target the Web-IO you should configure a reservation in the DHCP server so that the Web-IO can always be reached at the same address. The Ethernet address you need for this can be found on the white sticker on the unit.

```
5763x      [Model]
EN=00c03d003fa0  —— Ethernet address
OK xxxxxx
```
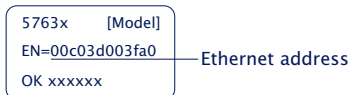
If in doubt ask the responsible network administrator.

**Networks without DHCP**

Install the WuTility program (download at *http://www.WuT.de*). If there is no Windows PC available, read the sub-section "Alternatives for IP address assignment" in the Appendix.

When starting WuTility the local subnet is searched and all W&T network components found are listed. Select your Web-IO and click on the IP Address icon. WuTility suggests the network parameters (subnet mask, gateway, DNS server), which also apply to the PC. If you want the Web-IO to  work in the same subnet as the PC, you need only adjust the IP address.

If under Address range you select *Any network*, you can also enter parameters that differ from your local network, e.g. for pre-configuring the Web-IO for a different network.

# Changing the set IP parameters

To change the IP address, subnet mask, gateway or DNS server after the fact, you can either use WuTility again or adjust the parameters in the browser at Basic settings » Network.

# 6. Basic Settings

The remaining configuration of the Web-IO takes place in the browser. For an address enter the IP address of the Web-IO. In the navigation tree click on *Login* and  select Administrator as the user. The default setting is for no password, and you only need top click on the Logion button to configure the Web-IO with Administrator rights.

## Configure IO channels

In *Basic settings » Inputs/Outputs* you can select whether the individual channels should be used as an input or output, and you can assign individual names. These names then replace the factory assigned designations *Channel n* in the visualization and any message texts.

## Datum / Uhrzeit

In *Date / Time* you can specify whether or not to cyclically calibrate with a time server. Date and time can also be set manually. Here you can also configure a time zone and daylight savings options.

## Language / Info

In addition to the choice of German and English you can also change other display elements including the logo.

## Password

Here you can specify the passwords for administrators and users.

*Please note that the same password may not be used for Administrator and User.*

If the Administrator password is no longer known, physical

access to the Web-IO is required in order to reset the passwords. See Emergency access in the Appendix to this manual.

# 7. Basic Applications

The Web-IO has a variety of communication options and supports various standard protocols. We recommend enabling only those communication paths that are actually needed in your application. This limits the possibility of undesirable outside access and tampering.

First we will show the three most used communication paths:

## Browser access

Access from the browser allows not only monitoring and operating of the IOs when appropriately logged in, but also configuration of the Web-IO.

The administrator has access to the entire configuration. User access (also password protected) allows the settings and actions pertaining to the IOs to be changed.

Without login only the states of the input and output can be monitored.

### HTTP or HTTPS
The factory setting is for browser access for HTTP using Port 80. To change access to HTTPS or to change the port, go to *Basic settings » Network* and then *Protocol* in the *Access for Web services* area. All other settings which pertain to the browser display can be made under *Web pages*.

### Hide menu tree
Once the configuration is completed, the browser display can be limited to IO access. To do this enable the *Hide menu tree* option under *Web pages >> Browser access*. At: *http://<URL/IP of the Web-IO>/index* the menu tree can be temporarily shown and then permanently enabled using the option above.

## IO access

The Web-IO offers two ready to use Web pages for access to the two channels:

## Home

The Home page provides an overview of all IOs and the configured actions. When appropriately logged in the channels configured as outputs can be set. This must first be enabled at *Web pages » Home*. By default this changing access is disabled.

Direct opening of the Home page without displaying the navigation tree at: *http://<URL/IP of the Web-IO>/home*

If *Hide menu tree* is enabled, a password entry field appears on the Home page. After clicking on the Apply button the outputs and counters can be operated until the Home page is exited again. Enabling the option *Web pages » Home > Save password for switching in browser* saves the password in the browser as a cookie, and after opening the Home page in the same browser operation is immediately enabled.

## My Web page

The Web page preloaded in the Web-IO offers a compact overview of the IO states.

At *Web pages » My Web page* you can replace the original Web page with one you created yourself.

*For this Web page to dynamically refresh the states of the channels, select* Communication paths » Web API > Allow HTTP requests. *Here you also specify whether the channels configured as outputs are allowed to be set using HTTP requests.*

Direct opening of your own Web page without displaying the navigation tree: *http://<URL/IP des Web-IO>/user*

For more details about programming your own Web pages, see the programming manual for the Web-IO. (Download at: http://*www.WuT.de* – simply enter the article number of your Web-IO in the search field and select Manual.)

# Sending e-mail

To send e-mail messages a few basic settings are necessary.

### Network parameters
To send via a mail server in the Internet, it is important that the network settings are correct. Check under *Basic settings » Network* to see whether the *Gateway* and *DNS Server* are correctly indicated.

### Mail server access
All mail server-specific settings can be made at *Communication paths » Mail*. The current most common authentication procedure is SSL/TLS. For more tips on the specific settings for the most common e-mail providers, see the Info area under *Mail*.

### Creating an e-mail message
To create a new e-mail message, click under *Actions* on the *Add* button. An input screen will appear for a new action.

Here you can specify the name of the action and what should be the initiator, e.g. the ON state of an input. A detailed description of the possibilities can be found in the *Actions* section.

For an action select *E-mail message*. In the corresponding input screen you can compose an individual e-mail message. Use the placeholders described below which can be replaced with the current IO states when sending the e-mail.

(x=Channel number)

| Placeholder | Description |
|---|---|
| **<Mx>** | Measurement value Channel 1 (commas) |

| Placeholder | Description |
|---|---|
| <m**x**> | Measurement value Channel 1 (decimal points) |
| <RC**x**> | Change rate in the last 5 minutes (commas) |
| <rc**x**> | Change rate in the last 5 minutes (decimal points) |
| <Z> | Date/time as string |
| <$y> | Year (####) |
| <$m> | Month (##) |
| <$d> | Day (##) |
| <$h> | Hour (##) |
| <$i> | Minute (##) |

## Box-to-Box

Box-to-Box mode connects two Web-IOs together over the network so that the outputs of the one follow the inputs of the other (A value on Input 0 of Web-IO A sets Output 0 of Web-IO B).

In Box-to-Box mode one Web-IO is configured as a master and the other as a slave. After a connection is successfully opened both Web-IOs work as equal partners and when correspondingly configured the switching signals travel in both directions.

# 8. Integration into Existing Systems

The Web-IO supports several common standards and protocols so that it can be easily integrated into many existing systems.

## MQTT

After enabling MQTT and configuration in *Communication paths » MQTT* the Web-IO supports two basic options:

1. Transferring the individual IO states as MQTT Topic to an MQTT Broker via MQTT Publish.

2. Setting the outputs based on Topic contents received via MQTT Subscribe.

Both cases are managed in the Web-IO as an alarm/message. A detailed description of the action principle used in the Web-IO can be found in the *Actions* section.

### Publishing IO states

To create a new MQTT Publish, click on Add under *Alarms/ Messages*. An input screen for a new action appears.

Here you can give a name to the action and what the initiator should be.

For example, you can specify Channel 1 as the initiator and a threshold value as the trigger.

As an action select *MQTT Publish*. In the following menu you enter the path for the topic to be written by the broker.

The text contents (payload) of the topic can be freely determined, whereby the placeholders described in the info text can be used.

**Setting the outputs via Subscribe**
Here also you must add a new alarm/message. As the initiator select *MQTT Subscribe*. Then enter the path for carrying the topic which contains the keyword for switching. As the action configure *Switch output » Switch output of this Web-IO*. Then specify what state the output should be switched to.

**Example:**

A device writes as a Topic the keyword 5 for the Broker indicated in the Web-IO over the path *wut/webio123/set0*. This path and the Topic are indicated for the Web-IO as an initiator under MQTT Subscribe. Setting the output to the value 5 is specified as the action.

The output is set each time a value is written.

**The Web-IO as MQTT gateway**
The flexibility offered by the Web-IO for message configuration means content-dependent Topics but also e-mails, SNMP traps or messages can be sent over different communication paths. See *Actions* section for more details.

# REST

REST (Representational State Transfer) is another Web-based communication path provided by the Web-IO.

Communication is via Web-IO-specific HTTP requests through the HTTP or HTTPS port entered under *Basic settings » Network » Access for Web-services*.

To be able to exchange REST data, access must first be enabled using *Communication paths » Rest*.

If REST access needs to be protected against unauthorized access, you can enable digest authentication. The requests must then be made as "admin" using the administrator password or as "operator" using the user password.

You can also specify whether the outputs are allowed to be set using REST.

### Read access

For read access REST uses the HTTP command GET.

The Web-IO supports three formats for replies to REST requests.

* JSON
* XML
* Text

The request can be used to determine in which format the reply comes. Using

```
http://<ip-adresse>/rest/json
```

for example means the entire process image of the Web-IO can be opened in JSON format. The reply then looks as follows:

```
{
  <info>:   {
    <request>:  </rest/json>,
    <time>:   <2018-01-01,15:15:58>,
    <ip>:   <192.168.0.123>,
    <devicename>: <WEBIO-07DD3A>
  },
  <iostate>: {
    <output>: [{
        <name>:   <Output 1>,
        <number>: 0,
        <unit>:   <V>,
        <value>:  0.0
      }, {
        <name>:   <Output 2>,
        <number>: 1,
        <unit>:   <V>,
        <value>:  0.0
      }]
  },
```

```
  <system>:{
    <time>:  <2018-01-01,15:15:58>,
    <diagnosis>:  [{
        <time>:  <01.01.2018 12:00:00>,
        <msg>: <Device state: OK>
      }],
    <diagarchive>:  [{
        <time>:  <01.01.2018 15:15:58>,
        <msg>: <Device state: OK>
      }]
  }
}
```

# OPC

The Web-IO is factory set for OPC operation. If you would like to use OPC, you must simply enable OPC access under *Communication paths.*

For your OPC client to communicate with the Web-IO the W&T OPC server must be installed. Access using OPC servers from third parties is not provided for.

In the OPC server select menu point *Devices » New I/O device.* Enter the IP address and password for your Web-IO and select the device type. Confirm with OK. Finally you must use *File » Save as active configuration* to apply the new entries.

# SNMP

Using SNMP both the IOs and the configuration of the Web-IO can be accessed. Which parameter, which status, which value can be called under which OID is stored in the private MIB, which can be downloaded from the *Web-IO Communication paths » SNMP* (alternately you can download from *http://www. WuT.de*).

The MIB can be conveniently viewed using one of the common MIB browsers. This is the quickest way to get an overview of the OID assignments.

All settings pertaining to SNMP can be made under *Communication paths » SNMP*. If the outputs need to be able to be set via SNMP, this must be enabled here.

## Opening an SNMP session

Read access is possible after enabling SNMP under *Communi-cation paths » SNMP* via SNMP-GET. For write/change access you must first log in to a session using the system password.

This is done using SNMP-SET via the OID, which you can find in the MIB branch of your Web-IO at

`wtWebGraphAnalog….SessCntrl » wtWebGraphAnalog….SessCntrlPassword`

To verify that a valid session is open, use a GET request to the OID

`wtWebGraphAnalog….SessCntrl » wtWebGraphAnalog….SessCntrlConfigMode`

(Reply 1 = valid session, 0 = no session.)

An existing session can be ended with a SET to the OID

`wtWebGraphAnalog….SessCntrl » wtWebGraphAnalog….SessCntrlLogout`

During an SNMP session login attempts from the browser are rejected.

## Access to the inputs and outputs

The input/output can be directly queried or set using SNMP Get instructions. The respective port is reached at the following paths:

<IP Address> 1.3.6.1.4.1.5040.1.2.**28**.1.**3**.1.1.1 = Output value 1 as Octet String with comma separators. (Read/Write)

<IP Address> 1.3.6.1.4.1.5040.1.2.**28**.1.**4**.1.1.1 = Output value

as integer value in thousandths, no comma separator. (Read/Write)

<IP Address> 1.3.6.1.4.1.5040.1.2.**28**.1.**8**.1.1.1 = Output value 1 as Octet String with decimal point separators. (Read/Write)

<IP Address> 1.3.6.1.4.1.5040.1.2.**28**.1.**3**.1.1.2 = Output value 2 as Octet String with comma separators. (Read/Write)

<IP Address> 1.3.6.1.4.1.5040.1.2.**28**.1.**4**.1.1.2 = Output value 2 as integer value in thousandths, no comma separator. (Read/Write)

<IP Address> 1.3.6.1.4.1.5040.1.2.**28**.1.**8**.1.1.2 = Output value 2 as Octet String with decimal point separators. (Read/Write)

The IDs for the various device versions are:

#57761: 1.3.6.1.4.1.5040.1.2.**28**...

#57762: 1.3.6.1.4.1.5040.1.2.**29**...

⚠️ *For the query enter the configured SNMP Read or Read/Write Community.*

## Modbus -TCP

Use *Communication path » Modbus-TCP* to enable the Web-IO for Modbus slave operation. Here you can also specify whether the outputs are permitted to be switched using Modbus-TCP.

The following tables show which function codes and register addresses are supported by the Web-IO.

All address information is in hex format.

There are various Modbus memory ranges for the Web-IO:

- Bit range (starting at Address 1000 or 1800)
- 16-bit range (starting at Address 2000),
- 32-bit range (starting at Address 5000, 6000 or 7000),

■ 8-bit range (Exception Status, no address).

Addressing is done in the bit-range, i.e. 1 bit requires an address. In the 16-bit and 32-bit range addressing is by the word (2 bytes).

The analog inputs are located in the 32-bit range starting at Address 5036 (example for 5766x: 5036 and 5038). The values are in percent with 3 places following the decimal point and are considered relative..

| Value | dec. | hex. | 4-20mA | 0-10V |
|-------|------|------|--------|-------|
| 0% | 0 | 0x0000 0000 | 4 | 0 |
| 1°% | 10 | 0x0000 000A | 4,0016 | 0.001 |
| 1% | 1,000. | 0x0000 03E8 | 4.16 | 0,1 |
| 10% | 10,000 | 0x0000 2710 | 5,6 | 1 |
| 65.5% | 65,535 | 0x0000 FFFF | 14.4856 | 6.55 |
| 65.5% | 65,536 | 0x0001 0000 | 14.4857 | 6.55 |
| 100% | 100,000 | 0x0001 86A0 | 20 | 10 |
| 120% | 120,000 | 0x0001 D4C0 | 23.2 | 12 |

The analog outputs are located in the 32-bit range starting at Address 5046 (example for 5776x: 5046 and 5047). The values (see above) are in percent with 3 places following the decimal point.

The alarms are located in the bit range starting at Address 1040 (example for 5776x: 1040 bis 1048), in the 16-bit range at 2004 and in the 32-bit range at Address 5004. The alarm trigger bits lie in the bit range starting at Address 1800.

The Exception-Status is located in the bit range starting at address 1060, in the 16-bit range at 200D (Low Byte). Alternately the Exception Status is read out using the function code 0x07.

The Configuration-Status is located in the bit range at Address 1068, in the 16-bit range at 200D (High Byte).

The Diagnostics-Status (number of errors) is located in the 16-bit range at 2006, in the 32-bit range at 504A.

The Diagnostics Status Bits lie in the 16-bit range starting at 2007, in the 32-bit range starting at 504C. #5766x allows 64 error messages.

The Device Identification is by serial number (starting at 6000) and Mac address (starting at 6004).

Available memory range, which the device provides for any Modbus client, lies in the 32-bit range starting at Address 7000.

| Memory-region | Definition | MB-Start-address [hex] | Read m. FC / Port access | Read m. FC / Block access | Write m. FC / Port access | Write m. FC / Block access | 5766x |
|---|---|---|---|---|---|---|---|
| Bit | Alarm- / Report-Status (1-8) | 1040 | 1,2 | - | - | - | 1-8 |
| Bit | Alarm- / Report-Status (9-16) | 1048 | 1,2 | - | - | - | 9 |
| Bit | Exception-Status | 1060 | 1,2 | - | - | - | Exc. 0-7 |
| Bit | Conf.-Status | 1068 | 1,2 | - | - | - | Conf.-St. 0-7 |
| Bit | Alarm-Trigger 1 | 1800 | 1,2 | - | 5 | 15 | 1 |
| Bit | Alarm-Trigger 2 | 1801 | 1,2 | - | 5 | 15 | 2 |
| Bit | Alarm-Trigger 3 | 1802 | 1,2 | - | 5 | 15 | 3 |
| Bit | Alarm-Trigger 4 | 1803 | 1,2 | - | 5 | 15 | 4 |
| Bit | Alarm-Trigger 5 | 1804 | 1,2 | - | 5 | 15 | 5 |
| Bit | Alarm-Trigger 6 | 1805 | 1,2 | - | 5 | 15 | 6 |
| Bit | Alarm-Trigger 7 | 1806 | 1,2 | - | 5 | 15 | 7 |
| Bit | Alarm-Trigger 8 | 1807 | 1,2 | - | 5 | 15 | 8 |
| 8Bit | Exception-Status | - | - | 7 | - | - | x |
| 16Bit | Alarm- / Report-Status (1-16) | 2004 | - | 3, 4 | - | - | 1-9 |
| 16Bit | Diagnose-Status (Anzahl Fehler) | 2006 | - | 3, 4 | - | 6, 16 | 0-x |
| 16Bit | Diagnose-Status (0-15) | 2007 | - | 3, 4 | - | - | 0-15 |
| 16Bit | Diagnose-Status (16-31) | 2008 | - | 3, 4 | - | - | 16-31 |
| 16Bit | Diagnose-Status (32-47) | 2009 | - | 3, 4 | - | - | 32-47 |
| 16Bit | Diagnose-Status (48-63) | 200A | - | 3, 4 | - | - | 48-63 |
| 16Bit | Diagnose-Status (64-79) | 200B | - | 3, 4 | - | - | 64 |
| 16Bit | Exception-Status (low byte) + Conf.-Status (high byte) | 200D | - | 3,4 | - | - | 0-7, 8-15 |

| Memory-region | Definition | MB-Start-address [hex] | Read m. FC / Port access | Read m. FC / Block access | Write m. FC / Port access | Write m. FC / Block access | 5766x |
|---|---|---|---|---|---|---|---|
| 32Bit | Alarm-/Report-St.(1-32) | 5004 | - | 3, 4 | - | - | 1-9 |
| 32Bit | AI 1 | 5036 | - | 3, 4 | - | - | 1 |
| 32Bit | AI 2 | 5038 | - | 3, 4 | - | - | 2 |
| 32Bit | AO 1 | 5046 | - | 3, 4 | - | 6,16 | 1 |
| 32Bit | AO 2 | 5048 | - | 3, 4 | - | 6,16 | 2 |
| 32Bit | Diagnose-Status (Error quantity) | 504A | - | 3, 4 | - | 6, 16 | 0-x |
| 32Bit | Diagnose-Status (0-31) | 504C | - | 3, 4 | - | - | 0-31 |
| 32Bit | Diagnose-Status (32-63) | 504E | - | 3, 4 | - | - | 32-63 |
| 32Bit | Diagnose-Status (64-95) | 5050 | - | 3, 4 | - | - | 64 |
| 32Bit | Serial number | 6000 | - | 3, 4 | - | - | OK-Nr. |
| 32Bit | Mac-address | 6004 | - | 3, 4 | - | - | Eth-Nr. |
| 32Bit | Memory 0 | 7000 | - | 3, 4 | - | 6, 16 | 0 |
| 32Bit | Memory 1 | 7002 | - | 3, 4 | - | 6, 16 | 1 |
| 32Bit | Memory 2 | 7004 | - | 3, 4 | - | 6, 16 | 2 |
| 32Bit | Memory 3 | 7006 | - | 3, 4 | - | 6, 16 | 3 |
| 32Bit | Memory 4 | 7008 | - | 3, 4 | - | 6, 16 | 4 |
| 32Bit | Memory 5 | 700A | - | 3, 4 | - | 6, 16 | 5 |
| 32Bit | Memory 6 | 700C | - | 3, 4 | - | 6, 16 | 6 |
| 32Bit | Memory 7 | 700E | - | 3, 4 | - | 6, 16 | 7 |
| 32Bit | Memory 8 | 7010 | - | 3, 4 | - | 6, 16 | 8 |
| 32Bit | Memory 9 | 7012 | - | 3, 4 | - | 6, 16 | 9 |
| 32Bit | Memory 10 | 7014 | - | 3, 4 | - | 6, 16 | 10 |
| 32Bit | Memory 11 | 7016 | - | 3, 4 | - | 6, 16 | 11 |
| 32Bit | Memory 12 | 7018 | - | 3, 4 | - | 6, 16 | 12 |
| 32Bit | Memory 13 | 701A | - | 3, 4 | - | 6, 16 | 13 |
| 32Bit | Memory 14 | 701C | - | 3, 4 | - | 6, 16 | 14 |
| 32Bit | Memory 15 | 701E | - | 3, 4 | - | 6, 16 | 15 |
| 32Bit | Memory 16 .. 31 | 7020 | - | 3, 4 | - | 6, 16 | 16-31 |

# 9. Alarms/Messages

Alarms/messages allow you to sent specific messages and set outputs. This is done based on defined IO states or other events.

Up to 12 messages can be stored and managed, whereby each message must have an individual name.

## Initiators

### Channel X
One of the two channels can be defined as the initiator. After entering the limit values, specify whether you want to monitor the range outside or within the configured limits. For monitoring within the limits you must enter a minimum and maximum value. To monitor outside the limits only a single limit needs to be entered.

### Interval timer
When configured appropriately the Web-IO can carry out actions at predefined times. Times are entered in cron format.

Valid characters are:

* \* Stands for all valid values in the respective input field (e.g. all minutes or all hours)
* \- Specifies a range of from..to (e.g. weekday "2-4" stands for Tuesday to Thursday, whereas entering "\*" initiates the timer on all weekdays).
* / Interval within the entered range (e.g. minute "0-45/2" initiates the timer in a range between the 0th and 45th minute every two minutes (0, 2, 4, 6 ,8, 10, ... , 44)).
* , Indicates an absolute value (e.g.: Minute "0, 15, 30" initiates the timer on the hour, at the quarter hour, and at the half hour.).

Example:

An action should be performed in the months of April to October every Monday at 8:00 a.m.

Minute:            0
Hour:              8
Day of month:      *
Month:             4-10
Day of the week:   1

### Device restart

When a restart initiates an action, the Web-IO distinguishes between two variants:

• Cold start

If the restart is initiated by a hardware intervention (applying or interrupting supply voltage or pressing the Reset key) the Web-IO considers this a cold start.

• Warm start

A warm start can be initiated from the Web page under Maintenance using the Restart button. In addition connecting to Port 8888 TCP and using the system password causes a restart if the Reset port is enabled.

### Sensor lost

If there is a sensor fault, i.e. no more values can be received, this can also be the initiator for a message.

## Actions

For actions which permit sending of alarm, message and other texts, placeholders can be used within the text which are replaced by actual contents, such as IO states, time etc. when an action is performed.

| Placeholder | Description |
|---|---|
| <M**x**> | Measurement value Channel 1 (commas) |
| <m**x**> | Measurement value Channel 1 (decimal points) |
| <RC**x**> | Change rate in the last 5 minutes (commas) |
| <rc**x**> | Change rate in the last 5 minutes (decimal points) |
| <Z> | Date/time as string |
| <$y> | Year (####) |
| <$m> | Month (##) |
| <$d> | Day (##) |
| <$h> | Hour (##) |
| <$i> | Minute (##) |

With text messages you can not only store the actual message sent, but also a Clear message. The Clear message is sent when the initiator for the action is no longer present – i.e. normal status has resumed. Message sending takes a variable amount of time depending on the protocol. If the initiating condition is only present for such a short time that the corresponding message could not even be sent, the Clear message is sent.

## E-mail message

The recipient, subject and contents of the e-mail can be freely configured.

To be able to send e-mail messages, access to the mail server must be configured and Mail enabled as the communication channel. All the necessary settings can be made under *Communication paths » Mail*. In the information area you can find the general access data for the most common e-mail providers.

### SNMP-Trap

The IP address and Host Name of the SNMP server as well as the message texts can be freely configured.

To be able to send SNMP traps, SNMP must be enabled under *Communication paths » SNMP*. All other parameters that can be set there are not relevant to sending SNMP traps.

### MQTT-Publish

The Web-IO can write any information as an MQTT Topic on a configurable path to an MQTT Broker.

This requires access to the MQTT Broker be configured under *Communication paths » MQTT.*

### HTTP-Request

Another possible action is sending of an HTTP request such as required by some devices such as cameras in order to trigger certain functions.

As the HTTP request enter the complete URL with all the parameters expected by the receiving device.

Format:
```
http://<Ip/Hostname>/<request>?Parameter1&Parameter2&ParameterN
```

For devices which require authentication with username and password, enable *Use authentication* and fill in the corresponding fields.

### TCP messages

When sending TCP messages the Web-IO works as a TCP client. When initiating an action it opens a TCP connection to the specified TCP server address on the specified port, transmits the message or clear text and then closes the connection. Any replies from the server are ignored and rejected.

**UDP messages**
To be able to send UDP messages, enable UDP Sockets in the UDP-Sockets ASCII-Mode area under *Communication pahts » Socket-API*.

When sending UDP messages the Web-IO acts as a UIDP peer. The message is transmitted in the form of a UDP datagram to the specified UDP peer address on the specified port. Any replies from the other end are ignored and rejected.

**Syslog messages**
IP Address and host name of the syslog server as well as the message texts can be freely configured.

To be able to send syslog messages, Syslog must be enabled under *Communication paths » Syslog*. All other parameters there are not relevant to sending syslog messages.

**FTP messages**
The Web-IO can save message texts to a file via FTP.

Under *Communication paths » FTP* FTP support must first be enabled and access to the FTP server configured.

The file name, message and clear texts can be formulated as desired.

The options are used to distinguish whether STOR can be used for each initiated action to completely overwrite the file or APPEND is used to append the message and clear texts continuously to the file.

**Switching outputs**
When switching outputs the Web-IO distinguishes between switching its own outputs of those of another Web-IO.

### Switching own outputs

When selecting an individual output it can be switched to ON or OFF. Another possibility is to change the existing state.

Alternately multiple outputs can be switched at the same time. You can specify for each selected output whether iot is turned ON or OFF.

### Switching the outputs of another Web-IO

Here again either one particular or multiple outputs can be switched.

Enter the IP address to specify which Web-IO to switch the outputs for. For the TCP port enter which one is set for the target Web-IO as access for the browser. If the target Web-IO is password protected, the password must be entered as well.

For the target Web-IO access for AJAX and HTTP requests must be enabled (*Communication paths » Web-API*) and the controlled outputs must be enabled for switching using HTTP requests.

The outputs on older model Web-IOs (#57661, #57662, #57630, #57631, #57634 and #57637) can also be switched. In this case specify the HTTP port of the Web-IO as the TCP port. In Output Mode Menu the outputs must be set to *HTTP*.

# 10. Access from Custom Applications

In addition to the numerous standardized access possibilities, the Web-IO also offers the option of accessing it from your own application.

This can be done using TCP/IP sockets from the common high-level languages. It is also possible however to use common Web techniques such as AJAX or PHP to communicate with the Web-IO.

## Access via TCP/IP sockets

The Web-IO offers three ways to access using TCP/IP sockets:

*   Command strings        ASCII
*   Binary structures        BINARY
*   HTTP requests        AJAX

### Command strings  ASCII

By exchanging simple command strings the inputs can be read and the outputs set.

Depending on the configuration the Web-IO operates in this most as a TCP server or UDP peer.

For setting the outputs you use the expression:

GET /outputaccessX?PW=<password>&State=<value>&

X: Number of the output: 1=Port 1, 2= Port 2

password: If an Admin password is assigned, it must be entered here in order to be able to set the output value. If no password is assigned, leave this place blank (...?PW=&...)

value: Here you enter the value you want to set on the

respective output. The unit of the value corresponds to the scale settings you configured.

To set a value of 50 on Channel 2 without an assigned password, use for example the expression:

GET /outputaccess2?PW=&State=50&

Response: <Header>;<Sensor-Name>;output2;<value> <unit>

example:
„http://<IP address>/outputaccess1?PW=&State=10.5&"
Response:
„<IP address>;WEBIO-xxxxxx;Sensor 1 4-20mA;output1;10,5 Unit"

Similar to setting the output, both input channels can also be polled using command strings.

The expression for polling the respective port is:

GET /SingleX

X: Number of the Inputs: 1=Port 1, 2= Port 2

Example:

10.40.42.44;WEBIO-046EE9;Sensor 1 0-20mA;14,300 mA

When entering the command string

GET /Single

without a port number the device outputs the values for both ports separated by semicolons:

10.40.42.44;WEBIO-046EE9;12,000 mA;5,000 mA

bzw.

12,000 mA;5,000 mA

**46**

## TCP server

To access the Web-IO as a TCP server using ASCII sockets, enable TCP ASCII-Sockets under *Communication paths » Socket-API*. Specify which server port the Web-IO should accept connections on. The Web-IO can simultaneously provide up to four TCP connections on the specified port – any additional connection attempt is rejected.

If the Web-IO does not receive a valid command within 30 seconds, it closes the connection and is then ready again for a new connection attempt. The Web-IO behaves in the same way when a defective or unknown command is received.

The inputs are generally read using polling. Event-driven evaluation is only possible if the input triggers are correspondingly configured.

## UDP peer

To access the Web-IO via UDP using ASCII sockets, enable UDP ASCII-Sockets under *Communication paths » Socket-API*. Specify on which local UDP port the Web-IO should accept datagrams.

Use Remote UDP-Port to specify at which UDP port of the requestor the replies should be sent from the Web-IO. Entering AUTO specifies that the replies go to the port which is entered in the receiving datagram as the sending port.

The inputs can only be read using polling. Event-driven evaluation is possible by adding a corresponding action (see *Actions* section).

## BINARY binary structures

The Web-IO specifies binary structures for various functions such as reading the inputs, setting the outputs etc. Access is exclusively by exchanging these structures.

In this mode the Web-IO can operate as a TCP client, TCP server or UDP peer. Access can be password protected.

Four binary accesses are available which can be configured independently of each other under *Communication paths » Socket-API*.

In TCP server mode only one client at a time can connect to the corresponding binary access. Any other connection attempt is rejected.

*A detailed description of the supported binary structures and other details about access using BINARY sockets can be found in the Web-IO programming manual. (Download at* http://www.WuT.de*). From the data sheet page of your Web-IO following the Manual link.*

## HTTP request
In addition to the traditional socket accesses, the Web-IO can also be accessed directly via HTTP using HTTP requests.

By default this access is disabled and must first be enabled at *Communication paths » Web-API*.

*A detailed description of the supported HTTP requests and other details about access using Web techniques such as AJAX or PHO can be found in the Web-IO programming manual. (Download at* http://www.WuT.de*). From the data sheet page of your Web-IO following the Manual link.*

# 11. Appendix

## Alternatives to IP address assignment

In cases where IP address assignment cannot be done using DHCP or the Wutility tool, the Web-IO offers another possibility:

### Assigning the IP address using the ARP command
*This method can be used if the Web-IO does not yet have an IP address, i.e. the entry is 0.0.0.0. Another prerequisite is that the Web-IO and the computer are in the same network segment.*

Read the Ethernet address of the Web-IO from the sticker on the side of the housing.

```
5763x      [Model]
EN=00c03d003fa0 ──── Ethernet address
OK xxxxxx
```
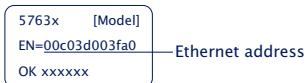
Now use the following command line to add a static entry to the ARP table of the computer:

```
arp -s [IP-Adresse]  [MAC-Adresse]
```

Example under Windows:
```
arp -s 10.40.72.15 00-C0-3-00-3F-A0
```

Example under SCO UNIX:
```
arp -s 10.40.72.15 00:C0:3D:00:3F:A0
```

Then start the Web browser and enter

```
http://<IP Address>
``` ein.

⚠️ *In Windows environments IP addresses must be entered without leading zeros.*

The Web-IO 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 Web page of the Web-IO is then loaded and all other settings can now be made using Web-based Management.

### Firmware updates
The Web-IO firmware is continually updated to always meet the new demands of growing networks.

*Current firmware for your Web-IO can be found at http://www. WuT.de by searching for the article number of the Web-IO and selecting the firmware.*

To load the firmware, you need a Windows PC with WuTility installed (included in the firmware archive) and unhindered network access to the Web-IO.

Start Wutility, select your Web-IO from the inventory list and click in the icon bar on Firmware. Select the corresponding UHD file. WuTility guides you through the update process.

*Do not interrupt either power or the network connection during the update.*

All the settings in the Web-IO remain unchanged and it should be immediately ready to use after the update.
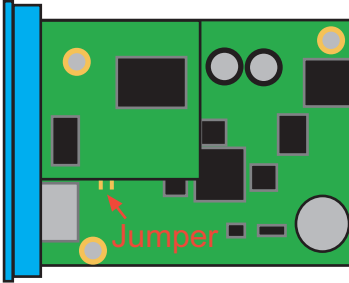
## Emergency access

In case you have forgotten the passwords for the Web-IO or simply want to restore the factory default settings, there are emergency access options depending on the model. In all cases you need physical access to the device.

### Models #57761, #57762
These models have a jumper on the motherboard that can be used to perform a complete reset to factory default settings.

First remove all connectors. Open the housing by slightly pressing with thumb and index finger on the housing body, which causes the housing walls to spread. Now the front panel together with the circuit board can be pulled out.

A jumper (normally open) can be found near the network socket.



Close the jumper and power up the Web-IO for approx. 30 seconds. The Web-IO is now configured with its factory default settings.

# 12. Technical Data

## Web-IO 4.0 Analog 0-20mA #57761

| Terminals and Indicators | |
|---|---|
| Current input passive | 0/4..20mA, 100Ohm |
| Current input active | 0/4..20mA, burden max. 500Ohm, active voltage Umax 14V / 40mA |
| Current output | 0/4..20mA, burden max. 500Ohm, supply 10V guaranteed |
| Analog in-/output | 8-pin terminal block<br>Conductor cross-section 0.2-1.3mm2<br>Strip to 6-8 mm |
| Current loop monitor | Short-circuit protected and open-loop detection |
| Network | 10/100BaseT autosensing<br>RJ45 |
| Galvanic isolation | Measuring inputs-network: min. 1500 V |
| Supply voltage | Power-over-Ethernet (PoE) or on screw terminal with DC 18V .. 48V (+/-10%) |
| Current consumption | typ. 100mA @24VDC, 60mA @48VDC<br>max. 120mA @24VDC, 70mA @48VDC<br>PoE Class 1 (0.44 – 3.84W) |
| Indicators | 1 LED Power<br>2 LEDs Network connection<br>4 LEDs Status and Error |
| **Measuring Unit** | |
| Resolution | Current input: 2.5µA |
| Measuring error | max. 0.5% FSR (Full Scale Range 0..20mA)<br>TA = 0-60°C |
| Sample rate | 15, 30 sec, 1, 5, 15, 60 min |
| Memory depth | min. 150 days, max. 99 years |
| Data logging (can be queried over the network or alarm triggering) | 2 / second |
| **Housing and other data** | |
| Housing | Compact plastic housing for DIN rail mount 105x22x-75mm (LxWxH) |
| Protection class | IP20 |
| Ambient temperature, storage | -40..+70°C |
| Ambient temperature, operating | 0 .. +60°C |

## Web-IO 4.0 Analog 0-10V #57762

| Terminals and Indicators | |
|---|---|
| Voltage input | 0..10V, 1MOhm |
| Voltage output | 0..10V, Imax 20mA<br>Switchable Sense line<br>Short-circuit protected |
| Analog in-/output | 8-pin terminal block<br>Conductor cross-section 0.2-1.3mm2<br>Strip to 6-8 mm |
| Network | 10/100BaseT autosensing<br>RJ45 |
| Galvanic isolation | Measuring inputs-network: min. 1500 V |
| Supply voltage | Power-over-Ethernet (PoE) or on screw terminal with<br>DC 18V .. 48V (+/-10%) |
| Current consumption | typ. 100mA @24VDC, 60mA @48VDC<br>max. 120mA @24VDC, 70mA @48VDC<br>PoE Class 1 (0.44 – 3.84W) |
| Indicators | 1 LED Power<br>2 LEDs Network connection<br>4 LEDs Status and Error |
| **Measuring Unit** | |
| Resolution | Voltage input: 1.25mV |
| Measuring error | max. 0.5% FSR (Full Scale Range 0..20mA)<br>TA = 0-60°C |
| Sample rate | 15, 30 sec, 1, 5, 15, 60 min |
| Memory depth | min. 150 days, max. 99 years |
| Data logging (can be queried over the network or alarm triggering) | 2 / second |
| **Housing and other data** | |
| Housing | Compact plastic housing for DIN rail mount 105x22x-75mm (LxWxH) |
| Protection class | IP20 |
| Ambient temperature, storage | -40..+70°C |
| Ambient temperature, operating | 0 .. +60°C |