Web-IO Programming - TCP/IP ASCII Sockets

The Web-IO provides simple socket access for TCP and UDP communication.

This access is disabled by default and must first be enabled via the menu path *Communication path >> Socket API*. For access via TCP, the Web-IO 4.0 is preset to server port 42280 (port 80 for older models). The Web-IO expects UDP datagrams on port 42279.

All commands begin with the keyword `GET` followed by a space. If parameters are included, the command string always ends with an `&`. Do not append a line break (CR LF).

Do not use the *writeln* method call, which is available in many high-level languages and automatically appends CR LF. In most cases, there is an equivalent method—*write* or *text* that transmits only the characters passed to it without appending anything.

Web-IO Digital

The following list shows all supported socket commands

Commands	Parameter	Description
`GET /input`**`x`**	`?PW=`**`password`**`&` The administrator or user password must be used instead of password. If no password has been assigned, `PW=&` is entered. (applies to all commands!)	Input status request **x** optionally specifies the no. of the input and can be between 0 and 16 depending on the model. The feedback of the Web IO is a string starting with `inputx`; followed by the input status: `ON` = signal at the input and `OFF` = no signal at the input If `x` is omitted completely, the Web IO returns a bit pattern corresponding to the input signals in hexadecimal notation.
`GET /counter`**`x`**	`?PW=`**`password`**`&`	Counter value request `x` can be a value between 0-11 and indicates the input. The feedback from the Web IO is a string beginning with `counterx;`. The counter value of the selected counter is appended in decimal notation.

Commands	Parameter	Description
GET /counter	?PW=**password**&	Requesting all counter values The feedback of the Web IO is a string beginning with counter;. The counter values are added in decimal notation separated by semicolons.
GET /outputx	?PW=**password**&	Output status request x optionally specifies the No. of the output and can be between 0 and 16 depending on the model. The feedback from the Web IO is a string starting with outputx; followed by the output status: ON = signal at the output and OFF = no signal at the output If x is omitted completely, the Web IO returns a bit pattern corresponding to the output signals in hexadecimal notation.
GET /outputaccessx	?PW=**password**& [Mask=**XXXX**&] State=**ON**/**OFF**/**YYYY**& [NA=ON&] ON: Output = 1, OFF: Output = 0, TOGGLE: Change of state **XXXX**: Hex value between 0000 and 0FFF according to the bits that are to be set **YYYY**: Hex value between 0000 und 0FFF according to the output bit pattern.	Set one or more outputs x can be a value between 0-11 and specifies the output to be set. The feedback of the Web IO is a string starting with „output;“ followed by a bit pattern corresponding to the output signals in hexadecimal notation. The specification of Mask is optional. If Mask is not sent, the outputaccess command applies to all outputs. With NA=ON it is optionally achieved that no response to the outputaccess command is sent by the Web IO.
GET /counterclear**x**	?PW=**password**& [Set=**value**&] **value**: Counter preset, value between 0 and 2 billion	Sets the counter value of a counter. If the Set parameter is not sent, the default is 0. x can be a value between 0-11 and specifies the input whose counter is to be reset. The response of the Web IO is a string beginning with counterx;. The new count of the selected counter is appended in decimal notation. If x is not specified, all counters are set.
GET /allout	?PW=**password**&	Collective request of the input, output states and all counter values. The Web IO responds with a string of the following structure: input;0xxx;output;0xxx;counter;n0;n1;n2,... 0xxx corresponds to the status of the inputs or outputs in hexadecimal notation. n0,n1 etc. contain the counter states in decimal notation.

Commands	Parameter	Description
GET /time	?PW=**password**&	Returns the system time of the Web IO. 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 passed with time.
GET /diagnosis	?PW=**password**&	Requests the status of the diagnostic memory. The Web IO returns: diagnosis;0000;00000000; 00000000;00000000 the four-digit value indicates the number of stored messages. With the three eight-digit hexadecimal values, each set bit represents one of the 92 possible messages.
GET /diagnosis**x**	?PW=**password**&	With x, the index for one of the currently stored messages is specified. The Web IO sends the corresponding message text as a return. x must not be greater than the number of the currently present messages.
GET /diaglist**x**	?PW=**password**&	Returns the messages for the individual message bits (max. 92)
GET /diagclear	?PW=**password**&	Clears message memory
GET /errorclear	?PW=**password**&	Clears load errors and releases the affected outputs.

Prepend the IP address and system name

In the menu branch *Communication path >> Socket API,* you can enable the option to prefix the IP address and system name.

In this case, the Web-IO's response is prefixed with the Web-IO's IP address and system name, separated by a semicolon.

```
<ip addresse>;<system name>;<parameter;<value>;.......
```

Example of a response to an input query with activation:

```
10.40.20.200;WEBIO-081500;input;0031
```

and without activation:

```
input;0031
```

Practical Application:

Read the complete process map

```
GET /allout?PW=<passwort>&
```

Example:

```
GET /allout?PW=blau&
```

Reply:

Web-IO Model	Reply
#57133, #57733	input;0;output;0;counter;0;0;0;0
#57150	input;0;output;0;counter;0;0
#57151	output;0
#57135, #57735	input;0;output;0;counter;0;0;0;0
#57153, #57753	input;0;output;0;counter;0;0;0;0
#57138, #57738	input;0000;output;0000;counter;0;0;0;0;0;0;0;0;0;0;0;0
#57730, #57630	input;0000;output;0000;counter;0;0;0;0;0;0;0;0;0;0;0;0
#57731, #57631	input;0000;output;0000;counter;0;0;0;0;0;0;0;0;0;0;0;0
#57732	input;0;output;0;counter;0
#57734, 57634	input;0000;output;0000;counter;0;0;0;0;0;0;0;0;0;0;0;0
#57736, #57756	input;0000;output;0000;counter;0;0;0;0;0;0;0;0;0;0;0;0
#57737, #57637	input;0;output;0;counter;0;0
#57832, #57838	output;00
#57650	input;0;output;0;counter;0;0;0;0;0;0

Read Input state

```
GET /input?PW=<passwort>&
```

Example:

```
GET /input?PW=blau&
```

Reply:

Web-IO Model	Reply
#57133, #57733	`input;0`
#57150	`input;0`
#57151	`not supported`
#57135, #57735	`input;0`
#57153, #57753	`input;0`
#57138, #57738	`input;0000`
#57730, #57630	`input;0000`
#57731, #57631	`input;0000`
#57732	`input;0`
#57734, 57634	`input;0000`
#57736, #57756	`input;0000`
#57737, #57637	`input;0`
#57832, #57838	`not supported`
#57650	`input;0`

Read Output state

```
GET /output?PW=<passwort>&
```

Example:

```
GET /output?PW=blau&
```

Reply:

Web-IO Model	Reply
#57133, #57733	`output;0`
#57150	`output;0`
#57151	`output;0`
#57135, #57735	`output;0`
#57153, #57753	`output;0`
#57138, #57738	`output;0000`
#57730, #57630	`output;0000`
#57731, #57631	`output;0000`
#57732	`output;0`
#57734, 57634	`output;0000`
#57736, #57756	`output;0000`
#57737, #57637	`output;0`
#57832, #57838	`output;00`
#57650	`output;0`

Set a single Output

```
GET /outputaccessx?PW=<passwort>&State=ON/OFF/TOGGLE&
```

Example:

```
GET /outputaccess0?PW=blau&State=ON&
```

Reply:

Web-IO Model	Reply
#57133, #57733	`output;1`
#57150	`output;1`
#57151	`output;1`
#57135, #57735	`output;1`
#57153, #57753	`output;1`
#57138, #57738	`output;0001`
#57730, #57630	`output;0001`
#57731, #57631	`output;0001`
#57732	`output;1`
#57734, 57634	`output;0001`
#57736, #57756	`output;0001`
#57737, #57637	`output;1`
#57832, #57838	`output;01`
#57650	`output;1`

Set multiple Outputs

```
GET /outputaccess?PW=<passwort>&State=ON/OFF/TOGGLE&
```

Example:

```
GET /outputaccess?PW=blau&State=0001&
```

Reply:

Web-IO Model	Reply
#57133, #57733	`output;1`
#57150	`output;1`
#57151	`output;1`
#57135, #57735	`output;1`
#57153, #57753	`output;1`
#57138, #57738	`output;0001`
#57730, #57630	`output;0001`
#57731, #57631	`output;0001`
#57732	`output;1`
#57734, 57634	`output;0001`
#57736, #57756	`output;0001`
#57737, #57637	`output;1`
#57832, #57838	`output;01`
#57650	`output;1`

Read all Counter

```
GET /counter?PW=<passwort>&
```

Example:

```
GET /counter?PW=blau&
```

Reply:

Web-IO Model	Reply
#57133, #57733	counter;0;0;0;0
#57150	counter;0;0
#57151	not supported
#57135, #57735	counter;0;0;0;0
#57153, #57753	counter;0;0;0;0
#57138, #57738	counter;0;0;0;0;0;0;0;0;0;0;0;0
#57730, #57630	counter;0;0;0;0;0;0;0;0;0;0;0;0
#57731, #57631	counter;0;0;0;0;0;0;0;0;0;0;0;0
#57732	counter;0
#57734, 57634	counter;0;0;0;0;0;0;0;0;0;0;0;0
#57736, #57756	counter;0;0;0;0;0;0;0;0;0;0;0;0
#57737, #57637	counter;0;0
#57832, #57838	not supported
#57650	counter;0;0;0;0;0;0

Reset all Counter

```
GET /counterclear?PW=<passwort>&
```

Example:

```
GET /counterclear?PW=blau&
```

Reply:

The format of the response is identical to that of the response to the `counter` command

Read a single Counter

```
GET /counterx?PW=<passwort>&
```

Example:

```
GET /counter0?PW=blau&
```

Reply:

Web-IO Model	Reply
#57133, #57733	counter0;0
#57150	counter0;0
#57151	not supported
#57135, #57735	counter0;0
#57153, #57753	counter0;0
#57138, #57738	counter0;0
#57730, #57630	counter0;0
#57731, #57631	counter0;0
#57732	counter0;0
#57734, 57634	counter0;0
#57736, #57756	counter0;0
#57737, #57637	counter0;0
#57832, #57838	not supported
#57650	counter0;0

Reset a single Counter

```
GET /counterclear0?PW=<passwort>&
```

Example:

```
GET /counterclear0?PW=blau&
```

Reply:

The format of the response is identical to that of the response to the `counterx` command

Web-IO Analog

Commands	Parameter	Description
GET /single**x**	none	Request for the current current or voltage values in mA or V. x optionally indicates the no. of the IO channel and can be 1 or 2. The feedback of the Web-IO is a string that shows the value with three decimal places and unit [N]N.NNN mA or [N] N.NNN V If x is omitted completely, the Web IO returns the values of both channels separated by semicolons.
GET /output**x**	none	gives the same result as GET /single**x** (even if IO channels work as input)
GET /outputaccessx	?PW=**password**& State=**N,NNN**& **N,NNN:** Strom- bzw. Spannungswert der am entsprechenden output eingestellt werden soll	Setting an output x can be 1 or 2 and indicates the output to be set. The feedback of the Web IO is a string in the format [N]N.NNN mA or [N] N.NNN V. and indicates the current value. Please note that the Web IO needs a few ms to set the desired value. Therefore, the value does not correspond to the desired value.
GET /time	?PW=**password**&	Returns the system time of the Web IO. 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 passed with time.
GET /diagnosis	?PW=**password**&	Requests the status of the diagnostic memory. The Web IO returns: diagnosis;0000;00000000; 00000000;00000000 the four-digit value indicates the number of stored messages. With the three eight-digit hexadecimal values, each set bit represents one of the 92 possible messages.
GET /diagnosis**x**	?PW=**password**&	With x, the index for one of the currently stored messages is specified. The Web IO sends the corresponding message text as a return. x must not be greater than the number of the currently present messages.
GET /diaglist**x**	?PW=**password**&	Returns the messages for the individual message bits (max. 92)
GET /diagclear	?PW=**password**&	Clears message memory

Practical Application:

As with Web-IO Digital, you can enable the option to prefix the response with the IP address and system name in the menu path *Communication pathes >> Socket API*

In this case, the IP address and system name are prepended to the actual response, separated by a semicolon.

Read analog value(s) using “single”

```
GET /single
```

Example:

```
GET /single
```

Reply:

Web-IO Model	Reply
#57161, #57761, #57642	`0,000 mA;0,000 mA`
#57162, #57762, #57643	`0,000 V;0,000 V`
#57641	`0,000 mA;0,000 V`

By appending the channel number (x = 1 or 2), you can also retrieve the current values individually:

```
GET /singlex
```

Example:

```
GET /single1
```

Reply:

Web-IO Model	Antwort
#57161, #57761, #57642	`0,000 mA;0,000 mA`
#57162, #57762, #57643	`0,000 V;0,000 V`
#57641 (Kanal 1)	`0,000 mA`
#57641 (Kanal 2)	`0,000 V`

Read analog value(s) using “output

```
GET /output?PW=<passwort>&
```

Not supported on models #57641, #57642 und #57643.

Example:

```
GET /output?PW=blau&
```

Reply:

Web-IO Model	Reply
#57161, #57761	`output;0,000 mA;0,000 mA`
#57162, #57762	`output;0,000 V;0,000 V`

By appending the channel number (x = 1 or 2), you can also retrieve the current values individually:

```
GET /output1?PW=blau&
```

Example:

```
GET /output1?PW=blau&
```

Reply:

Web-IO Model	Reply
#57161, #57761	`output1;0,000 mA`
#57162, #57762	`output1;0,000 V`

Set Outputs

Appending channel number x to the `outputaccess` command determines which channel will be set.

```
GET /outputaccessx?PW=<passwort>&State=0,000&
```

Not supported on models #57641, #57642 und #57643.

Example:

```
GET /outputaccessx?PW=blau&State=0,000&
```

Reply:

Web-IO Modell	Reply
#57161, #57761	`output1;0,000 mA`
#57162, #57762	`output1;0,000 V`

It takes some time for the voltage to stabilize, so the response does not always immediately return the value to be set. To verify the value, it is recommended to query it again using `single` or `output` at intervals of a few milliseconds.

Web-IO Climatic (Web-Therm., ...) and VOC

Commands	Parameter	Description
GET /single**x**	none	Request for the current current or voltage values in mA or V **x** optionally specifies the IO channel number and can be 1 through 8, depending on the model. The Web-IO's response is a string that returns the value with three decimal places and the unit [N]N.NNN mA or [N] N.NNN V If **x** is omitted entirely, the Web-IO returns the values of both channels separated by a semicolon.
GET /time	?PW=**password**&	Returns the Web-IO system time. 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 Web-IO system time to the value passed via time
GET /diagnosis	?PW=**password**&	Requests the status of the diagnostic memory. The Web-IO returns: diagnosis;0000;00000000; 00000000;00000000 The four-digit value indicates the number of stored messages. For the three eight-digit hexadecimal values, each set bit represents one of the 92 possible messages.
GET /diagnosis**x**	?PW=**password**&	**x** specifies the index for one of the currently stored messages. Web-IO returns the corresponding message text. **x** must not be greater than the number of currently available messages.
GET /diaglist**x**	?PW=**password**&	Returns the messages associated with the individual message bits (max. 92)
GET /diagclear	?PW=**password**&	Löscht den Meldungsspeicher

Practical Application:

As with the Web-IO Digital, you can enable the display of the IP address and system name in the menu path *Communication path >> Web API*.

In this case, the response is preceded by the IP address and the system name of the actual response, separated by a semicolon.

Read analog value(s) using “single”

```
GET /single
```

Example:

```
GET /single
```

Reply
#57107, #57172 Web-IO-Thermometer 2xPt100/Pt1000 (#57707,#57607,#57603)

```
25,0°C;25,0°C
```

Reply #57778 Web-IO-Thermometer 8xPt100/Pt1000 (#57708,#57608,#57604)

```
25,0°C;25,0°C;25,0°C;25,0°C;25,0°C;25,0°C;25,0°C;25,0°C
```

Reply #57113 Web-IO-ThermoHygrobarometer (#57713, #57613, #57612)

```
25,0°C;60,0%;995,0hPa
```

Reply #57114 Web-IO-Thermometer NTC (#57714, #57609)

```
25,0°C
```

Reply #57120 Web-IO-ThermoHygrometer (#57720, #57729)

```
25,0°C;60,0%
```

Reply
#57125, #57171 Web-IO-Thermometer Pt100/Pt1000 (#57725, #57715, #57610)

```
25,0°C
```

Reply #57726 Web-IO-Thermometer Relais (#57716, #57616)

`25,0°C;OFF` bzw. `25,0°C;ON`

Reply #57728 Web-Thermometer Air Quality

```
25,0°C;45,0%;475,0ppm
```