
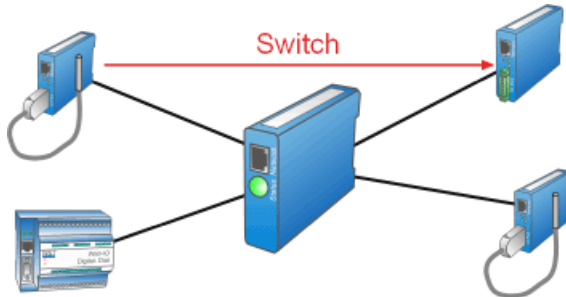



Application for pure.box:

Logically link Web-I/Os to each other

Product overview Application overview 

The example shown here illuminates the following function features of the pure.box:

- Time-controlled running of a PHP script
- Network communication via PHP

The states and measurement values recorded by multiple Web-I/Os (digital, analog, Thermograph etc.) will be used as a decision basis for switching digital outputs.

Application example

In order to monitor the temperature in a server room of a company a [Web-Thermo-Hygrobarograph PoE](#) is installed. The operating status of the air conditioner there can be accessed through a relay contact. This contact is brought to a [Web-I/O 2x Digital Input, 2x Digital Output PoE](#). The air conditioner also has its own room temperature monitor and turns itself on automatically as needed.

If the air conditioner stays off even though the temperature limit has been exceeded, a warning lamp is turned on at a janitor's display panel. The warning lamp is likewise connected to a [Web-I/O 2x Digital Input, 2x Digital Output PoE](#) and can therefore be turned on and off through the network.

Implementation

To implement this simple monitoring task the temperature in the server room and the operating status of the air conditioner must be checked. Based on the linked result the signal state of the warning light is set.

A W&T pure.box is used as the central intelligence. PNP support and the ability to run PHP script functions via CRON meets the requirements. Alternately any PC having the above properties can be used instead of the pure.box. The sample scripts for this application can be downloaded [here](#).

The file **webio_raw.php** contains standard functions which you can invoke from your PHP script. These functions can be used to retrieve and individually process states and measurement values from Web-I/Os. In the following we explain the available functions including the required invoking parameters and return values:

- **getDigitalIn(IP, Port, Channel, Password)**: Retrieves the current status of a channel from a digital Web-I/O. When invoking, the IP address, the HTTP port, the queried channel number and the password must be included. The return value is 0 or 1 and corresponds to the current status of the input. Example (no password assigned):
`getDigitalIn('10.4.27.10', 80, 2, '');`
- **getDigitalOut(IP, Port, Channel, Password)**: Like `getDigitalIn()`, but provides the status information for the queried output of the Web-I/O.
- **getCounter(IP, Port, Channel, Password)**: Retrieves the current counter state from the channel of a digital Web-I/O or Web-Counter. The parameters are IP address, HTTP port number, channel number and password. Returned is the counter state. Example: `getCounter('10.40.27.10', 80, 4, 'mein_passwort');`
- **setDigitalOut(IP, Port, Channel, Password, State)**: Sets the state of an individual output on a digital Web-I/O. In addition to the IP address, HTTP port, channel number and password the target state of the output must also be included. This is specified by 0 (= Output off) or 1 (= Output on). The returned value is the new switching state. The following example activates Output 6 on the Web-I/O having IP address 10.40.27.10: `setDigitalOut('10.40.27.10', 80, 6, 'mein_passwort', 1);`
- **getAnalogIO(IP, Port, Channel)**: Query in- and outputs on analog Web-I/Os and Web-Thermographs. Returns a float value. Example: `getAnalogIO('10.40.27.11', 80, 1);`
- **setAnalogOut(IP, port, channel, password, state)**: Sets the state of an individual output on an analog Web-I/O. In addition to the IP address, HTTP port, channel number and password the target state of the output must also be included. This is specified in numerical form (e.g. 2.54 or 7.23). The return value is the old analog value during the regulation process to the target value. The following example sets Output No. 1 on the Web-I/O with the IP address 10.40.27.11 to 3.5: `setDigitalOut('10.40.27.10', 80, 1, 'mein_passwort', 3.5);`

Assume the following example device constellation:

- Web-Thermograph in the server room with IP: 10.40.27.11, HTTP port: 80, measures the temperature on Channel 1
- Web-IO in server room with IP: 10.40.27.10, HTTP port: 80, signaling contact for the air conditioner on Input 0, no password assigned
- Web-IO at janitor's station with IP: 10.40.27.12, HTTP port: 80, warning light on Output 1, password: 'my_password'

You want the warning light to come on when the temperature exceeds 30°C and the signaling contact is inactive. The associated PHP script looks like this:

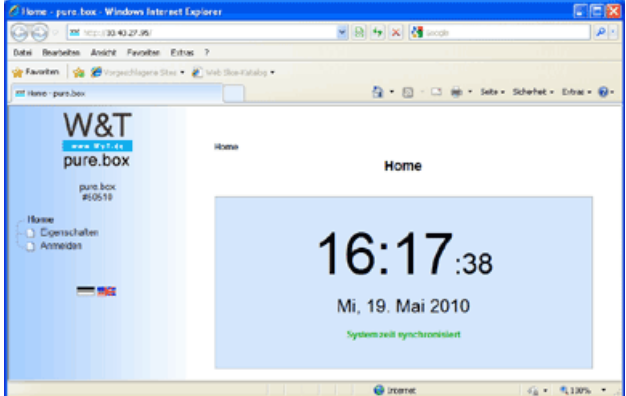


```
<?php
include './webio_raw.php';
$state = getDigitalIn('10.40.27.10', 80, 0, '');
$temperature = getAnalogIO('10.40.27.11', 80, 0);
$output = $state == 0 && $temperature > 30 ? 1 : 0;
setDigitalOut('10.40.27.12', 80, 1, 'mein_passwort', $output);
?>
```

First the include command is used to integrate the function library with the standard functions from the root directory. Only after integrating can the above described functions be used. In the variables **\$state** and **\$temperature** the signaling contact for the air conditioner and the current temperature in the server room are loaded. If the state is OFF and the temperature greater than 30°C, **\$output** is set to 1, otherwise to 0. Then the digital output to which the warning light at the janitor's station is connected is set to the determined state.

With the help of CRON controlled running of this script the monitoring can be performed at one-minute intervals.

Configuration of the pure.box

Save the file **webio_raw.php** via FTP or using the SMP signing directly in the root directory **userfiles** of the pure.box, and the file **webio_network.php**, after adapting it to your situation, in the program folder named **programs**. Then the following configuration steps are necessary in order to have the script run by the system once a minute:

<p>Open the Homepage of the pure.box by entering the IP address in the address line of your browser.</p>	
<p>Select Login in the menu tree and log in from the dialog box with administrator rights (user name: admin and your password).</p>	<p style="text-align: center;">Anmelden</p> <p style="text-align: center;">Passwort <input type="text"/></p> <p style="text-align: center;"><input type="button" value="Senden"/></p>
<p>Navigate to the page Home >> Configuration >> User files. In the file overview click on the folder named programs. The file webio_network.php is displayed.</p>	
<p>Clicking on the icon Edit file properties (left next to webio_network.php) opens the Edit dialog.</p>	

Activate **Time controlled** and ensure that in each of the following text fields a * is noted. As a result of this CRON definition the script is run at one-minute intervals.

Save the new settings... and you're done!

Zeitgesteuert	<input checked="" type="checkbox"/>	Komplette CRON-Syntax erlaubt
Minuten	<input type="text" value="*"/>	0-59
Stunden	<input type="text" value="*"/>	0-23
Monatstage	<input type="text" value="*"/>	1-31
Monate	<input type="text" value="*"/>	1-12
Wochentage	<input type="text" value="*"/>	0-6 (0 = Sonntag)

Summary

Simple monitoring tasks can be implemented in this way if the minimum monitoring cycle of one minute is compatible with the requirements. The advantage is that the individual signals can be distributed over the network, whereby local ties are eliminated.



We are available to you in person:

Wiesemann & Theis GmbH
Porschestra. 12
42279 Wuppertal
Phone: +49 202/2680-110 (Mon.-Fri. 8 a.m. to 5 p.m.)
Fax: +49 202/2680-265
info@wut.de

© Wiesemann & Theis GmbH, subject to mistakes and changes: Since we can make mistakes, none of our statements should be applied without verification. Please let us know of any errors or misunderstandings you find so that we can become aware of and eliminate them.

[Data Privacy](#)