

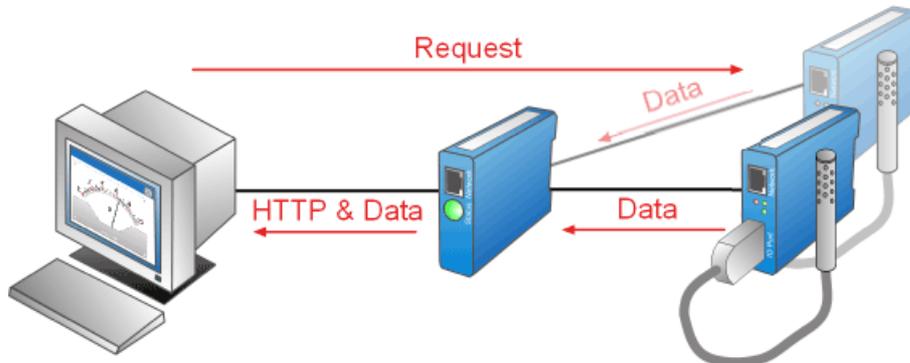
Application:

PHP data query

Product overview

Application overview

One webpage shows many devices...



Network communication between a browser and a terminal device using client-side Web techniques (e.g. JavaScript/AJAX, Java applet) is only possible if the terminal device itself provides the interrogating Web page. The security restrictions associated with these Web techniques restricts data exchange between the browser (Web page loaded by terminal device A) and other terminal devices (B, C, ...).

If multiple devices need to be accessed at the same time, e.g. in order to visualize the temperatures measured in multiple cool stores, central access with server-side run software (e.g. PHP) is required. If these prerequisites are present, the previously mentioned restrictions no longer apply. Initiating communication is now the task of the Web server which is authorized to establish connections with any terminal points.

Objective

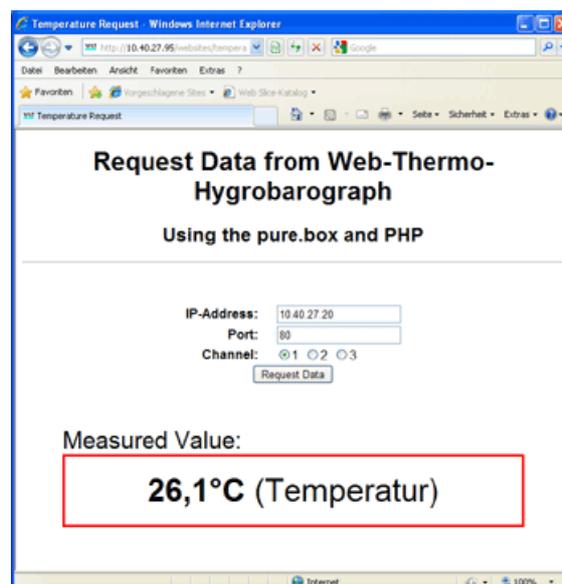
The present example describes how PHP is used to establish a socket connection to a data source. A measurement value is then requested through this connection which is finally displayed on the returned Web page. The required parameters (IP address, HTTP port and number of the measuring channel) come from a form and are sent when the page is opened via **Post**.

As a Web server with PHP support the [pure.box](#) is used. The data source with measuring points for temperature, relative humidity and air pressure is a [Web-Thermo-Hygrobarograph](#).

Only a PHP file is needed to implement this example, which is again invoked with the required parameter set for requesting new data. You can download the sample file [here](#).

Save the sample file **temperature_request.php** in the folder named **websites** on the pure.box. Use the following invoking URL from any browser:

`http://<IP-Adresse>/websites/temperature_request.php`



Source text

In the initiatory PHP area the variables are initialized. If the Web page is opened by entering the URL in the browser, no values for IP address, HTTP port and channel number via **Post** are sent. In this case the variables are filled with standard values. If the page is opened using the button **Request Data** on the Web page, the values entered by the user are sent and assigned to the variables.

```

<?php
if(isset($_POST["ip"]))
{
    $ip = $_POST["ip"];
    $port = $_POST["port"];
    $slot = $_POST["slot"];
}
else
{
    $ip = "";
    $port = "80";
    $slot = 1;
}
?>

```

In the **header** area the character set is specified and CSS used to determine the arrangement and design of central HTML components.

The form in the **body** is used for data entry by the user. The text fields **ip** and **port** are for the IP address and HTTP port of the requested Web-IOs. The radio buttons **slot** specify the requested channel. PHP is used to either fill these fields upon output with the standard values or the currently requested values. If the form is sent, the entered data are sent to the server by **Post**.

```

<form action="temperature_request.php" method="post">
<table align="center">
<tr>
<th>IP-Address:</th>
<td>
<input name="ip" type="text" value="<?php echo $ip; ?>" />
</td>
</tr>
<tr>
<th>Port:</th>
<td>
<input name="port" type="text" value="<?php echo $port; ?>" />
</td>
</tr>
<tr>
<th>Channel:</th>
<td>
<input <?php if($slot == 1) echo "checked"; ?> name="slot" type="radio" value="1" />1&nbsp;
<input <?php if($slot == 2) echo "checked"; ?> name="slot" type="radio" value="2" />2&nbsp;
<input <?php if($slot == 3) echo "checked"; ?> name="slot" type="radio" value="3" />3&nbsp;
</td>
</tr>
<tr>
<td align="center" colspan="2">
<input type="submit" value="Request Data" />
</td>
</tr>
</table>
</form>

```

Communication handling takes place in the PHP area within the DIV element **measurement**. If a parameter set is sent when opening, a socket connection to the specified IP address and port number is established. A timeout of 3s is defined for establishing the connection. In case of error the output text is **Connection Error**. The @ character before the **fsockopen** command also suppresses an error message generated by PHP. If the connection is successfully established the request **Get /single** followed by the desired channel number is used to request the measured value. The reply is then read in character by character until a 0-byte is received, indicating the end of the reply from the Web-IO. The character string composed at receipt is now converted into an array, whereby the semicolon is used as a separator for the individual elements. The last element of the array is the measured value including units, and the next to last element is the name of the channel. Both these are output.

If the page is opened with no parameter set, no communication is established, only three dashes (---) are output.

```
<?php
    if($ip)
    {
        $handle = @fsockopen($ip, $port, $errno, $errstr, 3);
        if (!$handle)
            echo "Connection Error!";
        else
        {
            $response = "";
            fwrite($handle, "GET /single" . $slot);
            do
            {
                $data = fgets($handle, 2);
                if(ord($data) != 0)
                    $response = $response . utf8_encode($data);
                else
                    break;
            }
            while(true);
            fclose($handle);
            $parts = explode(";", $response);
            echo "<b>" . $parts[count($parts)-1] . "</b> (" . $parts[count($parts)-2] . ")";
        }
    }
    else
        echo "---";
?>
```

Summary

The pure.box 2 with its PHP capable web server offers everything you need to communicate with other network components. You can access measurement values or data for various terminal devices from a central location and consolidate them on a web page. This gateway function is also useful for accessing a variety of terminal devices in private networks from the outside. All you need to do is set up NAT routing for the pure.box 2 instead of creating separate access for each of the terminal devices.



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](#)