

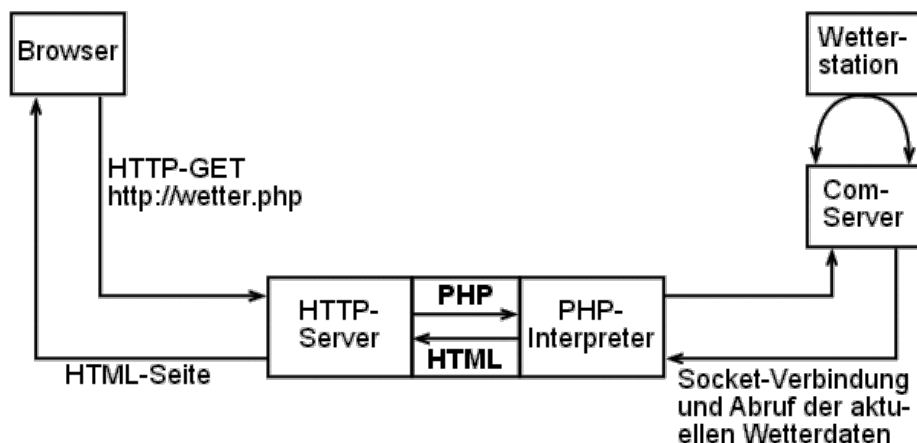
Applikation zum seriellen Com-Server:

Serielle Wetterstation per PHP ins Web

Häufig besteht die Anforderung auf Com-Server und die daran angeschlossenen seriellen Geräte mit Hilfe eines Standard-Web-Browsers zuzugreifen. Das heißt der Anwender gibt lediglich eine URL in das Adressfeld seines Browsers ein und erhält daraufhin die Webseite mit den aktuellen Daten des seriellen Gerätes. Das nachfolgende konkrete Beispiel beschreibt, wie mit Hilfe von PHP, Daten einer seriellen Wetterstation dynamisch in eine Webseite integriert werden.

Die Technik

PHP ist eine vom Syntax her an C angelehnte Interpretersprache, deren Quellcode üblicherweise in HTML-Seiten eingebunden wird. Statische Inhalte einer Webseite sind wie gewohnt in HTML formuliert. Veränderbare Bestandteile hingegen wie hier z.B. die Wetterdaten sind jedoch als PHP-Quellcode integriert. Ruft der Browser eine PHP-Seite ab, erfolgt zunächst ein Standard HTTP-GET an den Web-Server. Dieser erkennt an der Dateiendung *.php, daß die Seite PHP-Code enthält und reicht die Anforderung zunächst an den PHP-Interpreter weiter. Hier werden die PHP-Elemente ausgeführt und eine reine HTML-Seite an den Web-Server zurückgegeben der diese dann an den Browser sendet. Übertragen auf die konkrete Anwendung der Wetterstation gestaltet sich der Kommunikationsablauf wie skizziert.



Weitere Informationen über PHP inkl. der Verfügbarkeit für die verschiedenen Systemumgebungen, Downloadmöglichkeiten sowie Dokumentation finden Sie auf der Website <http://www.php.net>.

Das Programm

Nachfolgende finden Sie den Quelltext einer PHP-Seite wie sie auf dem Web-Server liegt und vom Browser abgerufen wird. Zur besseren Lesbarkeit sind alle PHP-Elemente rot bzw. blau dargestellt, wobei der eigentliche Programmcode rot und Kommentarzeilen blau sind. Der PHP-Interpreter erkennt für ihn relevante Bereiche an dem Start-Tag `<?php` und dem Ende-Tag `?>`. Standard-HTML-Code, der unverändert an den Browser zurückgegeben wird, ist hier schwarz dargestellt.

```

<?php
// Abfrage der Wetterstation "Davis Monitor II", angeschlossen an
// Com-Server 172.16.232.98.
// Max. Lebensdauer im Browser-Cache festlegen. 5 Minuten sind sinnvoll,
// da sich in dieser Zeit das Wetter nicht entscheidend ändert. Schlecht// wäre hingegen nach Aufruf des Lesezeichens
im Browser als erstes // Daten vom Vorabend zu bekommen.
header("Cache-Control: max-age=300");
// In diesem ersten PHP-Block werden die Wetterdaten im rohen
// Binärformat und mit Zeitstempel über eine Socketverbindung und den// Com-Server aus der Wetterstation gelesen.
Falls es dabei Probleme // gibt, wird stattdessen auf Daten aus einem Binär-Cache zurückge// griffen und ein
entsprechender Fehlerstring ausgefüllt.
// Der Pfadname der Binär-Cachedatei, muß in einem Verzeichnis liegen,
// für das der HTTP-Server Schreibrechte hat.
$strCacheFile = "/var/lib/wwwrun/weather.bin";
// IP-Adresse des Com-Servers

```

```

$IP = "172.16.232.98";
// Fehlermeldung, leer bedeutet "OK"
$strStatus = "";
$binData = "";
// Öffnen der TCP-Verbindung mit 2 Sekunden Timeout zum Com-Server.
$mySocket = @fsockopen( $IP, 8000, &$errno, &$errstr, 2 );
// Zweck der folgenden Aktion ist, im Fall "Connection refused *und*
// sehr alte Cache-Inhalt" unserem vermuteten Konkurrenten wenigstens// noch eine Chance zu geben, den Cache für
uns zu aktualisieren.
if( $errno == 111 and (time() - @filetime( $strCacheFile ) > 10) )
sleep( 1 );
if( $mySocket )
{
// Daten senden (1 Block Wetterdaten anfordern)
@fputs( $mySocket, "LOOP\xff\xff\r" );
// Es sollte mindestens ein Zeichen Antwort empfangen werden
socket_set_timeout( $mySocket, 2 );
$c = fgetc( $mySocket );
if( !$c )
{
$strStatus = "Wetterstation sendet keine Daten.";
// Den Com-Server resettet, da evtl. gar kein Endgerät ange-
// schlossen war, und es dann mehrere Minuten lang noch
// weiter versuchen würde, die Daten auszuliefern.
$dummySocket = @fsockopen( $IP, 9084 );
@fclose( $dummySocket );
}
else if( ord( $c ) == 6 )
{
// Lesen der 18 Byte langen Binärantwort Wetterstation über die
// Socket-Verbindung
$binData = fread( $mySocket, 18 );
$timeStamp = time();
// und diese in der Cache-Datei speichern
$myFile = @fopen( $strCacheFile, "wb" );
@fwrite( $myFile, $binData, 18 );
@fclose( $myFile );
}
else
{
$strStatus = sprintf( "Unerwartetes Quittungsbyte 0x%02x.", ord( $c ) );
// Schließen der Socket-Verbindung - würde sonst auch am Ende
// des Scripts automatisch erfolgen.fclose( $mySocket );
}
// Falls keine Daten empfangen wurden, diese der Cache-Datei entnehmen.
if( empty( $binData ) )
{
$myFile = @fopen( $strCacheFile, "rb" );
$binData = @fread( $myFile, 18 );
@fclose( $myFile );
$timeStamp = @filetime( $strCacheFile );
}
if( !$mySocket )
{
// Verbindung fehlgeschlagen, typische Fehlercodes sind dabei 111
// für RST (=Connection refused) und 110 für Timeout. "Connection
// Refused" melden wir nur, wenn außerdem die Daten der Cache-Datei
// ungewöhnlich alt sind, d. h. wenn der Com-Server anscheinend dauerhaft
// von jemand anders belegt ist.
if( $errno != 111 or time() - $timeStamp > 15 )
$strStatus = "TCP-Verbindung zur Wetterstation fehlgeschlagen ($errstr).";
}
?>
<HTML>
<HEAD>
...
<?php
// Der für die TCP-Verbindung und den Datenaustausch verantwortliche PHP-Code
// ist hiermit abgeschlossen. Ab hier folgt hauptsächlich für die
// Gestaltung der Seite verantwortlicher HTML-Code.
?>

```

Für Testzwecke oder zur weiteren Bearbeitung steht die komplette PHP-Datei hier als Download zur Verfügung: [↓
wetterdemo.php](#)

Die HTML-Antwort an den Browser

Nachdem der Interpreter den PHP-Code bearbeitet hat, wird dieser entfernt bzw. an den gewünschten Stellen durch HTML-Code ersetzt. Nachfolgend sehen Sie den HTML-Code, der aus dem o.a. Beispiel erzeugt und an den Browser gesendet wird. Die rot markierten Zeilen kennzeichnen die durch PHP dynamisch erzeugten Elemente.

```
<HEAD>
<TITLE>Davis Weather Monitor II</TITLE>
<META http-equiv="Refresh" content="60">
</HEAD>
<BODY bgcolor="maroon">
<TABLE align="center" width="90%" cellpadding="10">
<TR>
<TD bgcolor="white">
<H2>W&T Firmengebäude Wuppertal-Oberbarmen</H2>
Wetterdaten vom 26. Mär 2003, 16:48:09 (CET)
<BR>
<BR>
<TABLE width="90%" align="center">
<TR bgcolor="navy">
<TD width="50%"><FONT color="white">Messwert</FONT></TD>
<TD width="25%"><FONT color="white">innen</FONT></TD>
<TD width="25%"><FONT color="white">außen</FONT></TD>
</TR>
<TR bgcolor="silver">
<TD>Temperatur</TD><TD>23.4 °C</TD>
<TD>21.3 °C</TD>
</TR>
<TR bgcolor="aqua">
<TD>Luftfeuchtigkeit</TD>
<TD>21 %</TD>
<TD></TD>
</TR>
<TR bgcolor="silver">
<TD>Wind</TD>
<TD colspan="2">0.0 km/h aus N</TD>
</TR>
<TR bgcolor="aqua">
<TD>Luftdruck</TD>
<TD colspan="2">983 mbar</TD>
</TR>
</TABLE>
<BR>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```



Wir sind gerne persönlich für Sie da:

Wiesemann & Theis GmbH
Porschestra. 12
42279 Wuppertal
Tel.: 0202/2680-110 (Mo-Fr. 8-17 Uhr)
Fax: 0202/2680-265
info@wut.de

© Wiesemann & Theis GmbH, Irrtum und Änderungen vorbehalten: Da wir Fehler machen können, darf keine unserer Aussagen ungeprüft verwendet werden. Bitte melden Sie uns alle Ihnen bekannt gewordenen Irrtümer oder Missverständnisse, damit wir diese so schnell wie möglich erkennen und beseitigen können.

[Datenschutz](#)