


Applikation zum seriellen Com-Server:

PHP-Listing zum Artikel "Maschinen webfähig machen"

Produktübersicht Applikationsübersicht [↓ Download des PHP-Skripts als Textdatei](#)

```

<?php
// Abfrage der Wetterstation "Davis Monitor II", angeschlossen an
// Com-Server 172.16.232.98.

// Max. Lebensdauer im Browser-Cache in Sekunden festlegen. 5 Minuten
// sind sinnvoll, da sich in dieser Zeit das Wetter nicht entscheidend
// ändert, und auch eine Stunde wäre in der Sinne noch akzeptabel.
// Schlecht wäre hingegen, nach Aufruf des Besizers in Browser
// als erstes Daten vom Vorabend zu bekommen.
header: "Cache-Control: max-age=300" ;

// In diesem ersten PHP-Block werden die Wetterdaten im reinen BinärfORMAT
// über eine Socket-Verbindung zum Com-Server aus der Wetterstation gelesen
// und mit einem Zeitstempel versehen. Falls es dabei Probleme gibt, wird
// stattdessen auf Daten aus einem Browser-Cache zurückgegriffen und ein
// entsprechender Fehlerstrang ausgefüllt.

// Der Pfadname der Binär-Cachedatei, muß in einem Verzeichnis liegen, für
// das der HTTP-Server Schreibrechte hat.
$strCacheFile = "/var/lib/www/html/cache.bin";

// IP-Adresse des Com-Servers
$IP = "172.16.232.98";

// Initialisierung. Fehlermeldung-leer bedeutet "OK"
$strStatus = "";
$strData = "";

// Öffnen der TCP-Verbindung zum Com-Server, auf Port 8000,
// mit 2 Sekunden Timeout.
$mySocket = @fsockopen($IP, 8000, $errno, $errstr, 2);

// Wenn der folgenden Aktion ist, im Fall "Connection refused" (und
// "sehr alter Cache-Inhalt" unseren verarbeiteten Konkurrenten wenigstens
// noch eine Chance zu geben, den Cache für uns zu aktualisieren.
if ($errno == 111 and ($errno == $errno and $strCacheFile) > 10) {
    sleep(1);
}

if ($mySocket) {
    // Daten senden (1 Block Wetterdaten anfordern)
    fwrite($mySocket, "LOGSTAT$strIP");
    // Es sollte mindestens ein Zeichen Antwort empfangen werden
    $sock = @fread($mySocket, 2);
    $c = fgets($mySocket);
    if (!$c) {
        $strStatus = "Wetterstation sendet keine Daten.";
        // Den Com-Server resetten, da evtl. gar kein Faderan angeschlossen
        // war, und er dann mehrere Minuten lang noch weiter versuchen
        // würde, die Daten auszuliefern (Reset des Com-Servers durch An-
        // sprechen von Port 8084.
        $mySocket = @fsockopen($IP, 8084);
        fclose($mySocket);
    }
    else if (ord($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 {
        $strStatus = sprintf("Unerwartetes Quittungssymbol 0x%02x.", ord($c));
        // Schließen der Socket-Verbindung - wurde 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 (!$mySocket) {
    // Verbindung fehlgeschlagen, typische Fehlercodes sind dabei 111 für
    // ETC (=Connection refused) und 110 für Timeout. "Connection Refused"
    // melden wir nur, wenn außerdem die Daten der Cache-Datei
    // ungewöhnlich alt sind, d. h. wenn der Com-Server anscheinend
    // dauerhaft von jemand anders belegt ist.
    if ($errno != 110 and $timeStamp - $timeStamp > 15) {
        $strStatus = "TCP-Verbindung zur Wetterstation fehlgeschlagen !"
        . $errstr . " !";
    }
}

// Hier folgt der erste statische HTML-Teil für den Seitenaufbau:
<?
<HTML>
<HEAD>
<TITLE>Davis Weather Monitor II</TITLE>
<META http-equiv="Refresh" content="30">
</HEAD>
<BODY bgcolor="maroon">
<TABLE align="center" width="90%" cellpadding="10" cellspacing="10">
<TR>
<TD bgcolor="white">
<div style="text-align:center">

```

```

<?php $binData = fopen("binData.dat","r");
// Metadaten vom <?php setlocale( LC_TIME, "de DE" );
echo strftime( "%d. %b %Y, %X (%Z)", strtotime( 0 ) );

<BR>
<BR>
<TABLE width="80%" 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>
    <?php
      // In den folgenden drei PHP-Blöcken werden aus
      // den Binärfäden der Kennparameter die Temperatur-,
      // Feuchtigkeits-, Luftdruck und Windwerte extrahiert
      // und als HTML-Code eingefügt.
      //
      // Diese Schritte gelten nur für die hier verwendete
      // Wellensstation, dienen aber als Beispiel für das
      // Auswerten von Binärdaten an anderer Stelle.
      //
      // Dabei werden die als ASCII-Zeichen in einem String
      // vorliegenden Werte einzeln als 8-Bit-Binärwerte
      // interpretiert und weiter verarbeitet.
      // Byte 13...18 der Binärdaten werden in dieser Anwendung
      // ignoriert.
      //
      // Byte 1+2 sowie Byte 3+4 auslesen, jeweils zu 16-bit
      // Binärwerten zusammensetzen, in Celsius-Temperaturen
      // umrechnen und auf HTML-Seite darstellen.
      for( $i = 1; $i <= 2; $i += 2 )
      {
        $val = ord($binData[$i]) + (ord($binData[$i+1]) << 8);
        $val = ($val - 320)/18.0;
        printf( "<TD%.1F "</TD>"; $val );
      }
    </?php>

  <TR bgcolor="aqua">
    <TD>Luftfeuchtigkeit</TD>
    <?php
      // Byte 10 und 11 auslesen und als 2 Luftfeuchtigkeiten
      // auf HTML-Seite darstellen, falls Sensoren angeschlossen.
      for( $i = 10; $i <= 11; $i++ )
      {
        $val = ord($binData[$i]);
        if( $val == 0x80 )
          $val = "-"; // kein Sensor angeschlossen
        else
          $val .= " %";
        echo "<TD>".$val."</TD>";
      }
    </?php>

  <TR bgcolor="silver">
    <TD>Wind</TD>
    <?php
      // Byte 5 auslesen, in km/h umrechnen und als
      // Windgeschwindigkeit auf HTML-Seite ausgeben.
      $val = ord($binData[5]) * 1.609;
      printf( "<TD colspan='2'">%.1F km/h", $val );
      // Byte 6-7 auslesen, zu 16-bit Binärwert zusammensetzen
      // und aus Winkel in Grad die Windrichtung ermittelt und
      // auf HTML-Seite darstellen.
      $val = ord($binData[6]) + (ord($binData[7]) << 8);
      $val = intval($val / 22.5 + 0.5) % 16;
      $kompass = array( "N", "NNE", "NE", "ENE",
        "E", "ESE", "SE", "SSE",
        "S", "SSW", "SW", "WSW",
        "W", "WNW", "NW", "WNW" );
      echo " aus $kompass[ $val ]</TD>\n";
    </?php>

  <TR bgcolor="aqua">
    <TD>Luftdruck</TD>
    <TD colspan="2">
      <?php
        // Byte 8+9 auswerten, zu 16-bit-Binärwerten zusammensetzen, in
        // mbar Druck umrechnen und auf HTML Seite darstellen.
        $val = ord($binData[8]) + (ord($binData[9]) << 8);
        $val = intval( $val / 29.5 );
        echo "$val mbar</TD>\n";
      </?php>
    </TD>
  </TR>
</TABLE>
<BR>
</TD>
</TR>
<?php
// Eventuelle Fehlermeldungen werden auf der HTML-Seite dargestellt:
if( !empty( $strStatus ) )
{
  <TR bgcolor="red">
    <TD><FONT color="white">
      <?php echo $strStatus; ?>
    </FONT></TD>
  </TR>
</?php ! ?>
</BODY>
</HTML>

```

[↓ Download des PHP-Skripts als Textdatei](#)

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](#)