# Using WattPlot Data on Your Web Site with PHP

*by Sallye Holloway*

***Acknowledgements***

Thanks to Andrew Welch of Intallact for adding the System-by-Minute summary to WattPlot, an already awesome product, enabling us to get Mate data to use on the web. *www.wattplot.com*

Thanks to the PHP community for great online documentation and tons of examples to follow. *www.php.net/docs.php*

Thanks to the folks at maani for providing their cool XML/SWF tools to the public free of charge. *www.maani.us*

***Notes***

This was my first attempt at using PHP, and no doubt there are more elegant and more efficient ways of accomplishing things. I can't guarantee this will work for you, but I thought I'd document how I made woofpower.us in the spirit of those mentioned above. If you find it useful or have questions, please let me know! *sallye@woofpower.us*

# What You'll Need

The obvious things:

1. An Outback Power System
2. WattPlot that's hooked up to monitor the system
3. A web site configured to run PHP scripts
4. Some rudimentary programming skills

And depending upon how fancy you want to get, you may need some sort of graphics program to make nice dials, gauges or graphs.

## Step 1:

**Make sure the "system summary by minute" option is checked** in WattPlot's logging options. This will cause WattPlot to write **WPState.csv**, a comma-delimited file out once a minute that contains, you guessed it, the system summary. This is the data that we'll display on our web page. See the WattPlot documentation for what data is included.

## Step 2:

**Put the WPstate.csv in a location where your web site can access it.**

If your web site is hosted remotely, there are at least two options to achieve this:

First, you can configure the computer running WattPlot as a web server, and set WattPlot to save the log files in one of the accessible web folders.

> Windows XP Pro or later includes Microsoft's IIS web server.
> (Install from Control Panel > Add/Remove Programs > Windows Components.
> You'll need your Windows CD.)
>
> You can also download and install the free Apache Web Server.
> See http://httpd.apache.org/docs/1.3/windows.html

*If your web site is hosted remotely by a third party, they may restrict the ability for your web site to automatically open or upload remote files, and if so, this method may not work. This was my first plan, but I ended up doing the following:*

A simpler option is to use a batch program or macro program to regularly upload the WPstate.csv file to your remote web server. Here's an example of how to use the Windows XP built in ftp command:

> Use Notepad to make a file called WPupload.txt that contains the following lines:
>
> ```
> username
> password
> put WPState.csv
> quit
> ```
>
> Make a file called WPupload.bat that contains the following line:
>
> ```
> ftp -s:WPupload.txt ftp.mywebsite.com
> ```

*Username*, *password* and *ftp.mywebsite.com* should be replaced with the ones you use to log into your remote web site and upload files.

Put the two files into the same folder that WattPlot uses to store log files.

Then use the Windows Scheduler or other scheduler program to run WPupload.bat every 5 minutes, or any interval you choose. *Note: I ended up using a third party scheduler, since the Windows Scheduler kept choking for some reason.*

## Step 3:

**Get the data from the WPState.csv file and assign it to variables your web site can display.** wpdata.php is a sample PHP script that demonstrates this. You'll need to modify it to suit your particular system components and whatever statistics you are interested in displaying. Also, as WattPlot is updated in the future, you may need to adjust the order and specifics of the variables available in the WPState.csv log file. *I used the nifty freeware PHP editor CodeLobster (www.codelobster.com).*

### wpdata.php

```
<?php

/*
This code opens the WattPlot WPState.csv file and assigns the data to
variables for the main web site to display.  In this script, the csv file is in the
same directory as this php file.  If it is not, include the full path in the fopen
statement.  Some web hosting providers do not allow fopen to open a remote file.

Include this file in the default php file by using the following statement:
                        <?php require_once("wpdata.php"); ?>
*/


$msg_count = 0;              //clear message line count

if (is_readable("WPState.csv")) {

$fd = fopen("WPState.csv", "r");

while (!feof($fd)) {                    //loop until the end of file is reached
        $wp = fgetcsv($fd);             //get next line in file and put data in wp[ ] array

        /*
        Use the first item in each line to check which device the data is for
        and then load the data into that device's variables.
        Code can be expanded for multiple devices of the same type, as is shown
        with the inverter "FX" and charge controller "CC" sections  (I have two of each).
        */
        switch ($wp[0]) {

                case "STATUS":                  //this is a status line
                        $status_name = $wp[1];
                        $status_when = $wp[3];
                        break;

                case "MATE":                    //this is a mate line
                        $mate_number = $wp[1];
                        $mate_status = $wp[2];
                        $batt_voltage = $wp[3];
                        $batt_soc = $wp[4];
                        $batt_temp = $wp[5];
                        break;
```

```
case "FX":                              //this is an inverter line
        if ($wp[2] == "FX-1") {         //if this is inverter 1 load fx1 vars
                $fx1_id = $wp[2];
                $fx1_inverter_kWh = $wp[3];
                $fx1_charger_kWh = $wp[4];
                $fx1_buy_kWh = $wp[5];
                $fx1_sell_kWh = $wp[6];
                $fx1_aux_hrs = $wp[7];
                $fx1_batt_minVDC = $wp[8];
                $fx1_batt_maxVDC = $wp[9];
                $fx1_aux = $wp[10];
                $fx1_OPmode = $wp[11];
                $fx1_ACmode = $wp[12];
                $fx1_warn = $wp[13];
                $fx1_err = $wp[14];
                $fx1_ACin = $wp[17];
                $fx1_ACout = $wp[18];
                $fx1_invAmp = $wp[19];
                $fx1_chgAmp = $wp[20];
                $fx1_buyAmp = $wp[21];
                $fx1_sellAmp = $wp[22];
                $fx1_Alert = $wp[23];
                }
        elseif ($wp[2] == "FX-2") {  //else if this is inverter 2 load fx2 vars
                $fx2_id = $wp[2];
                $fx2_inverter_kWh = $wp[3];
                $fx2_charger_kWh = $wp[4];
                $fx2_buy_kWh = $wp[5];
                $fx2_sell_kWh = $wp[6];
                $fx2_aux_hrs = $wp[7];
                $fx2_batt_minVDC = $wp[8];
                $fx2_batt_maxVDC = $wp[9];
                $fx2_aux = $wp[10];
                $fx2_OPmode = $wp[11];
                $fx2_ACmode = $wp[12];
                $fx2_warn = $wp[13];
                $fx2_err = $wp[14];
                $fx2_ACin = $wp[17];
                $fx2_ACout = $wp[18];
                $fx2_invAmp = $wp[19];
                $fx2_chgAmp = $wp[20];
                $fx2_buyAmp = $wp[21];
                $fx2_sellAmp = $wp[22];
                $fx2_Alert = $wp[23];
                }
        break;
```

```
case "CC":                              //this is a charge controller line
        if ($wp[2] == "CC-3") {         //i have 2 charge controllers on ports 3,4
                $cc3_id = $wp[2];
                $cc3_charger_kWh = $wp[3];
                $cc3_panel_kWh = $wp[4];
                $cc3_daily_kWh = $wp[5];
                $cc3_float_hrs = $wp[6];
                $cc3_aux_hrs = $wp[7];
                $cc3_batt_minVDC = $wp[8];
                $cc3_batt_maxVDC = $wp[9];
                $cc3_pv_maxV = $wp[10];
                $cc3_OPmode = $wp[12];
                $cc3_chgAmp = $wp[13];
                $cc3_pvAmp = $wp[14];
                $cc3_pvV = $wp[15];
                $cc3_Alert = $wp[23];
                }
        elseif ($wp[2] == "CC-4") {
                $cc4_id = $wp[2];
                $cc4_charger_kWh = $wp[3];
                $cc4_panel_kWh = $wp[4];
                $cc4_daily_kWh = $wp[5];
                $cc4_float_hrs = $wp[6];
                $cc4_aux_hrs = $wp[7];
                $cc4_batt_minVDC = $wp[8];
                $cc4_batt_maxVDC = $wp[9];
                $cc4_pv_maxV = $wp[10];
                $cc4_OPmode = $wp[12];
                $cc4_chgAmp = $wp[13];
                $cc4_pvAmp = $wp[14];
                $cc4_pvV = $wp[15];
                $cc4_Alert = $wp[23];
                }
        break;

case "DC":    //this is a DC flexnet line, there can be only 1 per mate
        $dc_shunt1_kWh = $wp[3];        //i'm only using shunt1
        $dc_chargemet = $wp[6];
        $dc_aux_hrs = $wp[7];
        $dc_batt_minVDC = $wp[8];
        $dc_batt_maxVDC = $wp[9];
        $dc_batt_minSOC = $wp[10];
        $dc_shunt1_A = $wp[18];
        break;

case "MESSAGE":                         //a message line
        $msg_count = $msg_count + 1;    //increment the message count
        $msg[$msg_count] = $wp[1];
        break;
```

```
                case "\n";      //a line feed... just ignore
                        break;
                case "\r";      //a carraige return... just ignore
                        break;
                case NULL;
                        break;

                default:        //if none of the above, file could be corrupt so exit
                        exit("error: unrecognized data $wp[0]<br>");

        }

}
fclose($fd);            //finished reading the file

}       else {
        exit ("error: could not read data file");

}

/*set calculated variables here */

$pvmax_kW = 6900;
$batt_nom_V = 48;

$fx_inverter_kWh = $fx1_inverter_kWh + $fx2_inverter_kWh;
$fx_charger_kWh = $fx1_charger_kWh + $fx2_charger_kWh;
$fx_buy_kWh = $fx1_buy_kWh + $fx2_buy_kWh;
$fx_sell_kWh = $fx1_sell_kWh + $fx2_sell_kWh;
$fx_aux_hrs = $fx1_aux_hrs + $fx2_aux_hrs;

$fx_batt_minVDC = ($fx1_batt_minVDC <
$fx2_batt_minVDC)?$fx1_batt_minVDC:$fx2_batt_minVDC;

$fx_batt_maxVDC = ($fx1_batt_maxVDC >
$fx2_batt_maxVDC)?$fx1_batt_maxVDC:$fx2_batt_maxVDC;

$pv1kW = $cc3_pvV * $cc3_pvAmp * .001;
$pv2kW = $cc4_pvV * $cc4_pvAmp * .001;
$pvTkW = $pv1kW + $pv2kW;
$load_Amps = $fx1_invAmp - $fx1_sellAmp + $fx1_buyAmp - $fx1_chgAmp +
$fx2_invAmp - $fx2_sellAmp+ $fx2_buyAmp - $fx2_chgAmp;
```

```
$load = ($load_Amps * (($fx1_ACout + $fx2_ACout)/2))* .001;          //load in kW

$chgAmpDC = $cc3_chgAmp + $cc4_chgAmp;

$chgAmpAC = $fx1_chgAmp + $fx2_chgAmp;

$load_today_kWh = $fx_inverter_kWh - $fx_sell_kWh + $fx_buy_kWh -
        $fx_charger_kWh;

if ($fx_buy_kWh >= $fx_sell_kWh) {          //are we buying or selling power
        $sellflag = FALSE;
        $fx_today_kWh = $fx_buy_kWh - $fx_sell_kWh;
        }
else {
        $sellflag = TRUE;
        $fx_today_kWh = $fx_sell_kWh - $fx_buy_kWh;
        }

$chg_today_kWh = round(($fx_charger_kWh + $cc3_charger_kWh +
$cc4_charger_kWh),2);

$batt_temp = ((9/5)*$batt_temp)+32;      //change to farenheit


?>
```

wpdata.php end

## Step 4:

**Include wpdata.php in your default.php web page** with this statement:

```
<?php require_once("wpdata.php");  ?>
```

## Step 5:

**Use the variables to modify your web page** like this:

In this line, we display the $status_when variable:

```
<div class="smallbold">Status as of
<?php echo "  $status_when";      ?>
</div>
```

Once the PHP code is processed, if $status_when is "1/4/2010 2:14:52 PM", the following HTML code is sent to the browser:

```
<div class="smallbold">Status as of 1/4/2010 2:14:52 PM</div>
```

Another example:

```
<?php if ($sellflag == TRUE) : ?>          // check to see the status of $sellflag

        <p>I'm selling power!!  YAY!</p>  // if it's true, then do this

        ... other html code ...


<?php endif; ?>
```

See the online PHP documentation for help with using php in your web site.
*www.php.net/docs.php*

See the following page for a final text-only example: m.php, the WoofPower mobile home page.

```
<!-- parse WPState.csv and load data into variables -->
<?php require_once("wpdata.php");  ?>

Status as of<br>

<?php echo "$status_when "; ?>
[ <b>NOW</b> | <a href="m_today.php">TODAY</a> ]<br>
<hr>

<?php
if (($fx1_buyAmp + $fx2_buyAmp) >= ($fx1_sellAmp + $fx2_sellAmp)) {
        echo "<div class=\"buy\">BUYING $fx_buying_W W</div>";
} else {

        echo "<div class=\"sell\">SELLING $fx_selling_W W</div>";
}

echo "PV: $pvTkW kW<br>";
echo "LOAD: $load kW<br>";

echo "<div class=\"hdr\">Inverters:</div>";

echo "<b>Master:</b> $fx1_OPmode<br>$fx1_ACmode<br>";

if ($fx1_buyAmp > 0) {
        echo "<div class=\"buy\">Buying $fx1_buyAmp A</div>";
}
if ($fx1_sellAmp > 0){
        echo "<div class=\"sell\">Selling $fx1_sellAmp A</div>";
}

echo "<br><b>Slave:</b> $fx2_OPmode<br>$fx2_ACmode<br>";
if ($fx2_buyAmp > 0) {
        echo "<div class=\"buy\">Buying $fx2_buyAmp A</div>";
}
if ($fx2_sellAmp > 0) {
        echo "<div class=\"sell\">Selling $fx2_sellAmp A</div>";
}

// show warning & error count
        $warn = $fx1_warn + $fx2_warn;
        $errors = $fx1_err + $fx2_err;
        echo "<br>FX Warnings: <b class=\"warn\">$warn</b>";
        echo "<br>FX Errors: <b class=\"error\">$errors</b><br>";

echo "<div class=\"hdr\">Batteries:</div>";
echo "SOC: $batt_soc%<br>";
echo "$batt_voltage Volts<br>";
echo "$batt_temp F<br>";
echo "Charge Met: $dc_chargemet<br>";
echo "Charging from DC: $chgAmpDC Amps<br>";
echo "Charging from AC: $chgAmpAC Amps<br>";


?>
```

# Getting Fancy:

You can use the values from WPState.csv to configure the graphic components of your site. I've used the XML/SWF Gauges utility *(www.maani.us)* to move the needles on the gauges and update the text readouts on woofpower.us. Here are a few tips that may be of use if you decide to use this utility too.

Here's the basic structure:

index.php – your home page

dialsdef.php – the xml file that the swf file ultimately uses to determine what to draw on the screen

gauge.swf – the swf file from www.maani.com

wpdata.php – the php file discussed earlier in this document, modified to fit your system

Here's how it works:
1. index.php includes the wpdata.php file to read WPstate.csv and get the current Mate data as provided by WattPlot.

2. Index.php then passes variables of interest to dialsdef.php

3. Dialsdef.php calls gauge.swf to create the flash graphics.

When passing variables to the php file that defines your graphics, you must replace the ampersands with the hex equivalent **%26**. My script did not work when I replaced the equal signs as well as the ampersands as instructed in the XML/SWF docs. See the following snippet.

code snippet from index.php

```
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
        codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.
cab#version=6,0,0,0"
        WIDTH="600"
        HEIGHT="373"
        id="gauge">

<PARAM NAME="movie" VALUE="gauge.swf?xml_source=dialsdef.php<?php echo "?
Bv=$batt_voltage%26Bt=$batt_temp%26SOC=$batt_soc%26PV1=$pv1kW%26PV2=$
pv2kW%26PVT=$pvTkW%26LD=$load .... etc..."; ?>" />
<PARAM NAME="quality" VALUE="high" />
<PARAM NAME="bgcolor" VALUE="#000000" />
<param name="allowScriptAccess" value="sameDomain" />

<EMBED src="gauge.swf?xml_source=dialsdef.php<?php echo "?
Bv=$batt_voltage%26Bt=$batt_temp%26SOC=$batt_soc%26PV1=$pv1kW%26PV2=$
pv2kW%26PVT=$pvTkW%26LD=$load ... etc..."; ?>"
        quality="high"
        bgcolor="#000000"
        WIDTH="600"
        HEIGHT="373"
        NAME="gauge"
        allowScriptAccess="sameDomain"
        swLiveConnect="true"
        TYPE="application/x-shockwave-flash"
        PLUGINSPAGE="http://www.macromedia.com/go/getflashplayer">
</EMBED>
</OBJECT>
```

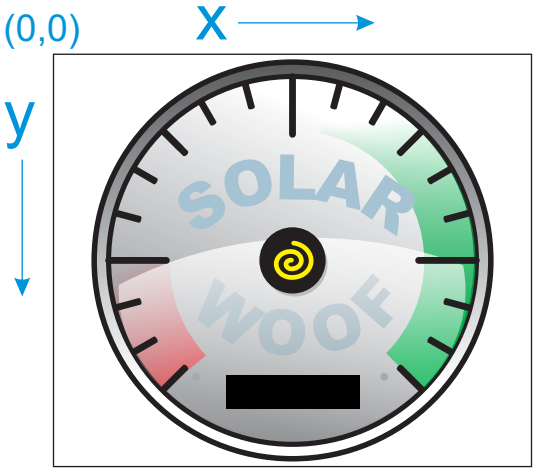
The corresponding code receives the passed variables in the dialsdef.php definition file:

code snippet from dialsdef.php

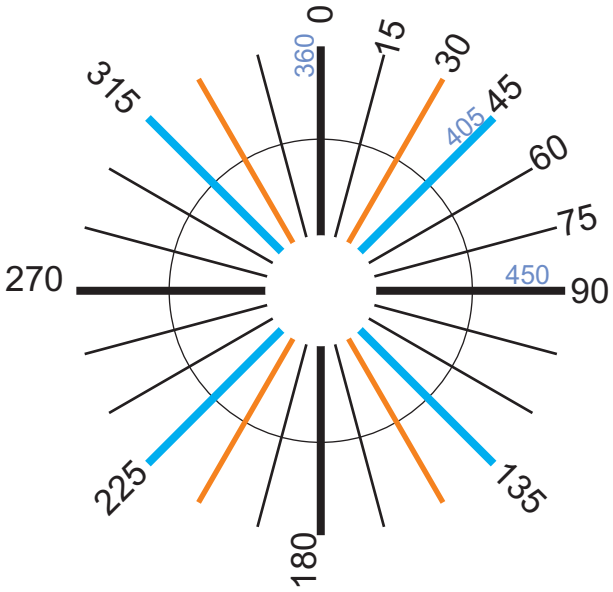
```
<gauge>
<image url='gaugebackground.jpg' />
        <?php
                //read the parameters from the query string passed from index.php
                $batt_voltage = $_GET['Bv'];
                $batt_temp = $_GET['Bt'];
                $batt_soc = $_GET['SOC'];
                $pv1kW = $_GET['PV1'];
                $pv2kW = $_GET['PV2'];
                $pvTkW = $_GET['PVT'];
                $load = $_GET['LD'];

.               ... etc ...
```

If you are drawing moving dials and text on top of a jpg image like I did, remember that the origin point of the image (0,0) is the upper left corner.




When using the rotate feature to rotate a dial on a gauge, 0 degrees is straight up, and the numbers wrap around like this:




So if your dial image looks like this:


You'd want the needle to move from 225 degrees to 495 degrees. The code in the XML definition file (*shown here as dialsdef.php*) would look something like:

## code snippet from dialsdef.php

```
<?php
        //figure where the pointer should be:  495-225 = 270 degree total span on my dials
        $deg_per_kW = (270/$max_gauge_val);       //degrees for each kW on the gauge

        //put the needle at the number of degrees for current value + the start angle
        $needle = ($current_kW * $deg_per_kW);
?>

<rotate x='405' y='81' start='225' span=<?php echo " '$needle' "; ?> step='3'
shadow_alpha='15' shake_frequency='3' shake_span='10'>
        <polygon fill_color='ff4400' fill_alpha='90' line_alpha='0'>
                <point x='402' y='70' />
                <point x='408' y='70' />
                <point x='406' y='30' />
                <point x='404' y='30' />
        </polygon>
        <rect x='402' y='92' width='7' height='5' fill_color='000000' fill_alpha='90'
line_alpha='50' />
        </rotate>

// show the current value on the gauge readout
 <text x='375' y='120' width='60' height='35'  align='center' size='16' color='00ff00' >
        <?php echo "$current_kW"; ?></text>
```

*define the needle pointing straight up*

That's pretty much it!  I created my own backgrounds, but photos, clip art or just using the XML/SWF libraries to draw gauges will also work fine.  If you have any questions, I'll do my best to answer.  Email me at sallye@woofpower.us.