

uPHP

This section contains details on how to use the Wattmon built in scripting language to customize it for any application. This page is an introduction to Programming the Wattmon and is of general interest to all users of the Wattmon, as it explains the architecture and design of the user interface software on the Wattmons.

Programming the Wattmon

“Programming” and “scripting” are used interchangeably throughout this documentation.

The Wattmon is built on a scripting language that is syntactically identical to PHP¹⁾ - we call it uPHP for Micro-PHP. It has a small but powerful subset of the functions and features found in the PHP language.

Why uPHP?

First perhaps it would be good to explain a bit about how and why we chose PHP. The goal of the Wattmon is to provide a highly flexible platform that can be customized for pretty much any application imaginable, keeping in mind very low memory availability. It became clear that a scripting language would be required to implement this, since having custom firmwares for various applications would be very tedious. The question then became: what would be the *best* scripting language that is well known today and can be deployed on a microcontroller with such limited resource? The choices are actually very small. Most modern scripting languages require quite a bit of RAM. So, we decided to roll our own scripting language from scratch and base it off a well known language.

The Wattmon interface is primarily browser based, and generates HTML pages on the fly from uPHP scripts. The methodology of HTML-PHP integration is well understood by most programmers and hence it allows you to become more productive quickly.

Challenges

The biggest challenge with the scripting language is its memory limitation. The [WattmonPRO](#) and [WattmonMINI](#) processors have a total of 128KB (yes, Kilobytes!) of RAM available to run a multitasking OS with a TCP/IP and USB stack, modbus polling engine, scheduler and more. For these platforms during normal operation scripts have a maximum of about 30 KB of memory for variables, which means that as a programmer you need to be very conscious of memory allocation. We have introduced some special non-PHP compatible variable types to help with this. The [indexed_array\(\)](#) function lets you create arrays of a very specific type (int, byte, etc) which will reduce the amount of resources compared to hash arrays created with the [array\(\)](#) function.

The [WattmonMEGA](#) processor has a total of 512KB of RAM (4 times as much as for the PRO and the MINI). Memory limitations are much less of a challenge when scripting for the MEGA.

Benefits

Having a scripting language that lets you modify and edit your program remotely is a very powerful thing. It lets you provide remote support to your clients and fix issues without needing to visit the site, and also makes your applications future-proof as you can at any time update the code base and integrate new features or protocols.

Further Information on uPHP

Of interest to those who want to dig deeper and advance into making tweaks and changes to the system:

[uPHP Language Basics](#)

[uPHP Function Reference](#)

¹⁾

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. PHP stands for “PHP: Hypertext Preprocessor.” This confuses many people because the first word of the acronym is the acronym (this type of acronym is called a recursive acronym.)

From:

<http://wattmon.com/dokuwiki/> - **Wattmon Documentation Wiki**

Permanent link:

<http://wattmon.com/dokuwiki/uphp?rev=1535793189>

Last update: **2021/09/13 05:56**

