

# uPHP Reference

uPHP functions have an identical syntax to PHP functions in most cases. Below is a list of all the functions that have been implemented.

Certain additional functions are available as WattmonOS include files. For a list of these see [library\\_functions](#).

Click on the function name for further details:

FUNCTION NAME	PARAMETER(S)	RETURN	DESCRIPTION
<a href="#">adc_read</a>	<a href="#">int</a> <a href="#">channel</a>	<a href="#">int</a> ADC value	Read an onboard ADC <a href="#">channel</a>
<a href="#">aes_decrypt</a>	<a href="#">string</a> text, <a href="#">int</a> length, <a href="#">string</a> key, <a href="#">string</a> iv	string with data	Return an AES-decrypted string
<a href="#">aes_encrypt</a>	<a href="#">string</a> text, <a href="#">int</a> length, <a href="#">string</a> key, <a href="#">string</a> iv	string with data	Return an AES-encrypted string
<a href="#">array</a>	<a href="#">mixed</a> values ...	<a href="#">array</a>	Create an <a href="#">array</a> , with optional <a href="#">values</a>
<a href="#">array_key</a>	<a href="#">array</a> , <a href="#">int</a> index	<a href="#">string</a> key	Return the key for an <a href="#">array</a> <a href="#">index</a>
<a href="#">array_keys</a>	<a href="#">array</a> with key/value pairs	<a href="#">array</a> of keys	Return keys for an <a href="#">array</a> that has key/value pairs
<a href="#">array_resize</a>	<a href="#">array</a> indexed array, <a href="#">int</a> length	none	Resize indexed <a href="#">array</a>
<a href="#">base64_decode</a>	<a href="#">string</a> base64	<a href="#">string</a> decoded or <a href="#">int</a> 0	Decode a base64-encoded <a href="#">string</a>
<a href="#">base64_encode</a>	<a href="#">string</a> to encode	<a href="#">string</a> base64 or <a href="#">int</a> 0	Return the base64-encoded version of a <a href="#">string</a>
<a href="#">call_user_func</a>	<a href="#">string</a> function_name, <a href="#">mixed</a> parameters ...	<a href="#">mixed</a> result	Call a user defined function with optional <a href="#">parameters</a>
<a href="#">charat</a>	<a href="#">string</a> , <a href="#">int</a> index	<a href="#">int</a> ASCII code	Return the ASCII code for a character in a <a href="#">string</a> at an <a href="#">index</a>
<a href="#">chdir</a>	<a href="#">string</a> directory	<a href="#">int</a> 0=OK	Change the current directory
<a href="#">chr</a>	<a href="#">int</a> code	<a href="#">string</a> 1 character	Return the character for an ASCII <a href="#">code</a>
<a href="#">clear_watchdog</a>			Clear the software watchdog timer
<a href="#">cos</a>	<a href="#">number</a> <a href="#">radian_angle</a>	<a href="#">float</a> cosine	Return cosine of a <a href="#">radian_angle</a>
<a href="#">debug</a>	<a href="#">string</a> output		Print to debug output
<a href="#">debugout</a>	<a href="#">int</a> 0 or 1		Enable or disable debug messages
<a href="#">die</a>			Kill the script
<a href="#">disk_free_space</a>	<a href="#">int</a> drive	<a href="#">int</a> KiloBytes	Return free space on drive
<a href="#">disk_status</a>	<a href="#">int</a> drive	<a href="#">int</a> Status	Return mount status of drive
<a href="#">disk_total_space</a>	<a href="#">int</a> drive	<a href="#">int</a> KiloBytes	Return total space on drive
<a href="#">download</a>	<a href="#">string</a> url	<a href="#">int</a> res	Download a file in the background

FUNCTION NAME	PARAMETER(S)	RETURN	DESCRIPTION
<a href="#">download_info</a>		<a href="#">array</a> info	Get info about ongoing download
<a href="#">download_state</a>		<a href="#">int</a> status	Get state of download
<a href="#">error_reporting</a>	<a href="#">int</a> verbosity		Set the debug output level
<a href="#">ereg</a>	<a href="#">string</a> pattern, <a href="#">string</a> content [, & <a href="#">array</a> matches]	<a href="#">int</a>	Perform a regex operation
<a href="#">exec</a>	<a href="#">string</a> script, <a href="#">int</a> delay		Run a <a href="#">script</a> with an optional <a href="#">delay</a>
<a href="#">exec_action</a>	<a href="#">mixed</a> action	<a href="#">int</a> 1=OK	Triggers a manually executable <a href="#">action</a> by id or name
<a href="#">explode</a>	<a href="#">string</a> , <a href="#">string</a> delimiter	<a href="#">array</a>	Turn a <a href="#">string</a> into an <a href="#">array</a>
<a href="#">f485open</a>	<a href="#">int</a> baud, <a href="#">int</a> parity	<a href="#">int</a> handle or 0	Open the RS-485 port at the specified <a href="#">baud</a> rate and <a href="#">parity</a>
<a href="#">fclose</a>	<a href="#">int</a> handle		Close a file, stream or socket
<a href="#">feof</a>	<a href="#">int</a> handle	<a href="#">int</a> 1 or 0	Test if no more data is available in a file, stream or socket
<a href="#">fgets</a>	<a href="#">int</a> handle, <a href="#">int</a> size	<a href="#">string</a> or <a href="#">int</a> -1	Return a single line from a file, stream or socket, with optional <a href="#">size</a> limit
<a href="#">file_exists</a>	<a href="#">string</a> filename	<a href="#">int</a> 1 or 0	Check if a file exists
<a href="#">filesize</a>	<a href="#">string</a> filename or <a href="#">int</a> handle	<a href="#">int</a> bytes	Return the size of a file, or the number of unread bytes in a stream or socket
<a href="#">findfirst</a>	<a href="#">string</a> pattern, <a href="#">int</a> attributes	<a href="#">array</a> first file found	Start searching the current folder for files matching a <a href="#">pattern</a> and <a href="#">attributes</a>
<a href="#">findnext</a>		<a href="#">array</a> next file found	Return next matching file information (after a findfirst)
<a href="#">firmwareupdate</a>			Initiate a firmware update sequence and reboot the device
<a href="#">floatval</a>	<a href="#">mixed</a> value	<a href="#">float</a> value or <a href="#">int</a> 0/1	Return the <a href="#">float</a> value of a <a href="#">number</a> or <a href="#">string</a>
<a href="#">flush</a>	<a href="#">socket</a> socket to flush		Flush current output or socket to the browser
<a href="#">fopen</a>	<a href="#">string</a> filename, <a href="#">string</a> mode	<a href="#">int</a> handle or 0	Open a file for reading or writing
<a href="#">fread</a>	<a href="#">int</a> handle, <a href="#">int</a> bytes	<a href="#">string</a> or <a href="#">int</a> 0	Read <a href="#">bytes</a> from a file, stream or socket
<a href="#">fread_unpack</a>	<a href="#">int</a> handle, <a href="#">string</a> format, <a href="#">int</a> count, <a href="#">int</a> interval	<a href="#">number</a>	Write contents of an indexed array to a file in binary
<a href="#">freemem</a>		<a href="#">int</a> bytes	Return free memory space
<a href="#">freestack</a>		<a href="#">int</a> bytes	Return free stack space
<a href="#">fseek</a>	<a href="#">int</a> handle, <a href="#">int</a> offset, <a href="#">int</a> whence		Position the file pointer in an open file
<a href="#">fseropen</a>	<a href="#">int</a> baud, <a href="#">int</a> blocking, <a href="#">int</a> invert, <a href="#">int</a> parity	<a href="#">int</a> handle or 0	Open the serial port at the specified <a href="#">baud</a> rate with optional parameters

FUNCTION NAME	PARAMETER(S)	RETURN	DESCRIPTION
<a href="#">fsockopen</a>	<a href="#">string</a> host, <a href="#">int</a> port, <a href="#">int</a> timeout	<a href="#">int</a> handle or 0	Open an internet socket connection with optional <a href="#">timeout</a>
<a href="#">ftell</a>	<a href="#">int</a> handle	<a href="#">int</a> position	Return the current position of a file read/write pointer
<a href="#">ftp_command</a>	<a href="#">string</a> result	<a href="#">string</a> command]	Send an FTP command
<a href="#">ftp_close</a>	<a href="#">int</a> result		Close an active connection with FTP server
<a href="#">ftp_download</a>	<a href="#">int</a> result	<a href="#">string</a> remote_file, <a href="#">string</a> local_file, [ <a href="#">int</a> position]	Initiate a download of a remote file
<a href="#">ftp_error</a>	<a href="#">int</a> result		Get last FTP response code
<a href="#">ftp_is_busy</a>	<a href="#">int</a> result		Check if the FTP engine is busy
<a href="#">ftp_is_connected</a>	<a href="#">int</a> result		Check if the FTP connection is active
<a href="#">ftp_is_connecting</a>	<a href="#">int</a> result		Check if the FTP connection is in the process of connecting
<a href="#">ftp_list</a>	<a href="#">int</a> result	<a href="#">string</a> folder, <a href="#">string</a> output_file	Lists a folder on the FTP server and outputs the result to the specified file
<a href="#">ftp_open</a>	<a href="#">int</a> result	<a href="#">string</a> host, <a href="#">int</a> port, <a href="#">string</a> username, <a href="#">string</a> password, <a href="#">int</a> keepalive, <a href="#">int</a> ignore_reply	Open a connection to an FTP server
<a href="#">ftp_size</a>	<a href="#">int</a> size	<a href="#">string</a> filename	Get the file size of a file on the FTP server
<a href="#">ftp_status</a>	<a href="#">array</a> status		
<a href="#">ftp_upload</a>	<a href="#">int</a> result	<a href="#">string</a> remote_file, <a href="#">string</a> local_file, [ <a href="#">int</a> position]	Initiate an upload of a file
<a href="#">function_exists</a>	<a href="#">string</a> function_name	<a href="#">int</a> 1 or 0	Check if a function exists (custom or native)
<a href="#">fwrite</a>	<a href="#">int</a> handle, <a href="#">mixed</a> data, <a href="#">int</a> length	<a href="#">int</a> bytes written or -1	Write <a href="#">data</a> to a file, stream or socket
<a href="#">fwrite_pack</a>	<a href="#">int</a> handle, <a href="#">array</a> data, <a href="#">int</a> length	<a href="#">number</a>	Write contents of an indexed array to a file in binary
<a href="#">get3gstat</a>		<a href="#">array</a>	Get cellular data connection status information
<a href="#">getcwd</a>		<a href="#">string</a> path	Get the current directory
<a href="#">getethstat</a>		<a href="#">array</a>	Get Ethernet connection status information
<a href="#">getmac</a>		<a href="#">string</a> MAC	Get the Wattmon's MAC address
<a href="#">gettype</a>	<a href="#">any</a> variable	<a href="#">string</a> type	Get a variable type as a string
<a href="#">getusbstat</a>		<a href="#">array</a>	Get USB host status information
<a href="#">getwifistat</a>		<a href="#">array</a>	Get WIFI status information
<a href="#">header</a>	<a href="#">string</a> header_data		Add to HTTP header

FUNCTION NAME	PARAMETER(S)	RETURN	DESCRIPTION
<a href="#">hash_hmac</a>	<a href="#">string</a> <a href="#">algorithm</a> , <a href="#">string</a> <a href="#">data</a> , <a href="#">string</a> <a href="#">key</a>	<a href="#">string</a> converted	Generate a keyed hash value using the HMAC method
<a href="#">htmlspecialchars</a>	<a href="#">string</a> <a href="#">data</a>	<a href="#">string</a> converted	Convert special characters for display in HTML
<a href="#">ieee754toint</a>	<a href="#">float</a> <a href="#">value</a>	<a href="#">int</a> representation	Convert a <a href="#">float</a> <a href="#">value</a> to an IEEE-754 encoded <a href="#">integer</a> (32 bit)
<a href="#">implode</a>	<a href="#">array</a> , <a href="#">string</a> <a href="#">delimiter</a>	<a href="#">string</a>	Turn an <a href="#">array</a> into a <a href="#">string</a>
<a href="#">include</a>	<a href="#">string</a> <a href="#">filename</a>		Include a file within the current script at the current location
<a href="#">indexed_array</a>	<a href="#">int</a> <a href="#">type</a> , <a href="#">int</a> <a href="#">size</a>	<a href="#">array</a>	Create an <a href="#">array</a> of a specific <a href="#">type</a> and <a href="#">size</a>
<a href="#">ini_get</a>	<a href="#">string</a> <a href="#">filename</a> , <a href="#">string</a> <a href="#">section</a> , <a href="#">string</a> <a href="#">key</a> , <a href="#">mixed</a> <a href="#">default</a>	<a href="#">mixed</a> value	Get a value from an INI file
<a href="#">ini_get_array</a>	<a href="#">string</a> <a href="#">filename</a> , <a href="#">string</a> <a href="#">section</a>	<a href="#">array</a>	Get a group of parameters from an INI file as an <a href="#">array</a>
<a href="#">ini_put_array</a>	<a href="#">string</a> <a href="#">filename</a> , <a href="#">array</a> <a href="#">data</a> , <a href="#">string</a> <a href="#">section</a>		Write a group of parameters to an INI file from an <a href="#">array</a>
<a href="#">ini_set</a>	<a href="#">string</a> <a href="#">filename</a> , <a href="#">string</a> <a href="#">section</a> , <a href="#">string</a> <a href="#">key</a> , <a href="#">mixed</a> <a href="#">value</a>	<a href="#">int</a> 1=OK	Set a <a href="#">value</a> in an INI file
<a href="#">init_watchdog</a>	<a href="#">int</a> <a href="#">interval</a>		Initialize the software watchdog timer
<a href="#">inttoieee754</a>	<a href="#">int</a> <a href="#">representation</a>	<a href="#">float</a> <a href="#">value</a>	Convert an IEEE-754 encoded <a href="#">integer</a> <a href="#">representation</a> (32 bit) to a <a href="#">float</a>
<a href="#">intval</a>	<a href="#">mixed</a> <a href="#">value</a>	<a href="#">int</a> <a href="#">value</a>	Return the <a href="#">integer</a> value of a <a href="#">number</a> or <a href="#">string</a>
<a href="#">is_array</a>	<a href="#">mixed</a> <a href="#">variable</a>	<a href="#">int</a> 1 or 0	Check if a <a href="#">variable</a> is an <a href="#">array</a>
<a href="#">is_float</a>	<a href="#">mixed</a> <a href="#">variable</a>	<a href="#">int</a> 1 or 0	Check if a <a href="#">variable</a> is a <a href="#">float</a>
<a href="#">is_int</a>	<a href="#">mixed</a> <a href="#">variable</a>	<a href="#">int</a> 1 or 0	Check if a <a href="#">variable</a> is an <a href="#">integer</a>
<a href="#">is_numeric</a>	<a href="#">mixed</a> <a href="#">value</a>	<a href="#">int</a> 1 or 0	Check if a <a href="#">value</a> is numeric ( <a href="#">int</a> , <a href="#">float</a> or numeric <a href="#">string</a> )
<a href="#">is_string</a>	<a href="#">mixed</a> <a href="#">variable</a>	<a href="#">int</a> 1 or 0	Check if a <a href="#">variable</a> is a <a href="#">string</a>
<a href="#">isset</a>	<a href="#">mixed</a> <a href="#">variable</a>	<a href="#">int</a> 1 or 0	Check if a <a href="#">variable</a> exists
<a href="#">json_encode</a>	<a href="#">array</a> , <a href="#">int</a> <a href="#">method</a>	<a href="#">string</a>	JSON encode an <a href="#">array</a> into a <a href="#">string</a> , with optional <a href="#">method</a>
<a href="#">ln</a>	<a href="#">number</a> <a href="#">number</a>	<a href="#">float</a> $\log_e$	Return the natural logarithm of a <a href="#">number</a>
<a href="#">log</a>	<a href="#">string</a> <a href="#">output</a> , <a href="#">string</a> <a href="#">file</a>		Print to the System Log (or optional <a href="#">file</a> )
<a href="#">log10</a>	<a href="#">number</a> <a href="#">number</a>	<a href="#">float</a> $\log_{10}$	Return the base 10 logarithm of a <a href="#">number</a>
<a href="#">mail</a>	<a href="#">string</a> <a href="#">recipient</a> , <a href="#">string</a> <a href="#">subject</a> , <a href="#">string</a> <a href="#">body</a>	<a href="#">int</a> 0 or SMTP error code	Send an email [deprecated]
<a href="#">max_execution_time</a>	<a href="#">int</a> <a href="#">seconds</a>		Set the maximum execution time for the current script

FUNCTION NAME	PARAMETER(S)	RETURN	DESCRIPTION
<a href="#">mb_add_dev</a>	<a href="#">int id</a> , <a href="#">int type</a> , <a href="#">string name</a> , <a href="#">int poll_interval</a> , <a href="#">int status</a> , <a href="#">int bus</a>	<a href="#">int</a> 0=OK	Add a device to the list of polled devices
<a href="#">mb_delete_dev</a>	<a href="#">int id</a>	<a href="#">int</a> 1=OK	Delete a device from the list of active devices
<a href="#">mb_get_dev_by_id</a>	<a href="#">int id</a>	<a href="#">array</a>	Return modbus device details by <a href="#">id</a>
<a href="#">mb_get_dev_by_index</a>	<a href="#">int index</a>	<a href="#">array</a>	Return modbus device details by <a href="#">index</a>
<a href="#">mb_get_dev_by_name</a>	<a href="#">string name</a>	<a href="#">array</a>	Return modbus device details by <a href="#">name</a>
<a href="#">mb_get_dev_info</a>	<a href="#">int type</a>	<a href="#">array</a>	Return modbus device details by <a href="#">type</a>
<a href="#">mb_get_role_array</a>		<a href="#">array</a>	Return an <a href="#">array</a> of all roles and their values
<a href="#">mb_get_status_by_role</a>	<a href="#">int role</a>	<a href="#">int</a> 1=OK	Return status of the device attached to the <a href="#">role</a>
<a href="#">mb_get_val_by_role</a>	<a href="#">int role</a>	<a href="#">number</a>	Return value of the <a href="#">role</a>
<a href="#">mb_num_devices</a>		<a href="#">int</a>	Return number of devices on the modbus
<a href="#">mb_queue_command</a>	<a href="#">mixed values ...</a>	<a href="#">array</a> of numbers	Queue a sequence of characters to the RS-485 bus and get but ignore the reply
<a href="#">mb_scan_complete</a>		<a href="#">int</a> 1=complete, 0=ongoing	Check to see if a modbus scan has completed
<a href="#">mb_scan_percent</a>		<a href="#">number</a> percent completed	Return scan percentage completed
<a href="#">mb_send_command</a>	<a href="#">mixed values ...</a>	<a href="#">array</a> of numbers	Send a sequence of characters to the RS-485 bus and get a reply
<a href="#">mb_set_dev_var</a>	<a href="#">string name</a> or <a href="#">int id</a> , <a href="#">string variable</a> , <a href="#">mixed value</a>	<a href="#">int</a> 1=OK	Set a <a href="#">variable</a> on a modbus device
<a href="#">mb_set_val_by_role</a>	<a href="#">int role</a> , <a href="#">number value</a>	<a href="#">int</a> 1=OK	Set a <a href="#">role value</a> on a modbus device
<a href="#">mb_start_scan</a>	<a href="#">int start</a> , <a href="#">int end</a>		Initiate an automatic scan of the modbus
<a href="#">md5</a>	<a href="#">string input</a>	<a href="#">string</a> 32 characters	Calculate the MD5 hash of a <a href="#">string</a>
<a href="#">md5_file</a>	<a href="#">string filename</a>	<a href="#">string</a> 32 characters	Calculate the MD5 hash of a file
<a href="#">mem_dump</a>			Write the current memory map to /dump.txt
<a href="#">mem_usage</a>			Write memory usage to standard output
<a href="#">microtime</a>		<a href="#">int</a> ms	Return the number of milliseconds since boot
<a href="#">mkdir</a>	<a href="#">string pathname</a>	<a href="#">int</a> 0 or error code	Make a directory

FUNCTION NAME	PARAMETER(S)	RETURN	DESCRIPTION
<a href="#">mktime</a>	<a href="#">int</a> hour, <a href="#">int</a> minute, <a href="#">int</a> second, <a href="#">int</a> month, <a href="#">int</a> day, <a href="#">int</a> year	<a href="#">int</a> seconds	Return the Linux Timestamp for a given date and time
<a href="#">mqtt_disconnect</a>			Disconnect MQTT connection
<a href="#">mqtt_publish</a>	<a href="#">string</a> channel, <a href="#">string</a> content	<a href="#">int</a> 1 for success or 0 for error	Publish a message to an MQTT server
<a href="#">mqtt_subscribe</a>	<a href="#">string</a> channel, <a href="#">string</a> callback	<a href="#">int</a> 1 for success or 0 for error	Subscribe to a channel on an MQTT server
<a href="#">mqttstat</a>		<a href="#">array</a> array with connection status	Get MQTT Connection status
<a href="#">net_disable3g</a>			Disable 3G support for the dongle
<a href="#">net_enable3g</a>			Enable 3G support for the dongle
<a href="#">netstat</a>		<a href="#">array</a>	Get Ethernet information
<a href="#">number_format</a>	<a href="#">mixed</a> number, <a href="#">int</a> digits	<a href="#">string</a> formatted	Return the <a href="#">string</a> value of a <a href="#">number</a> formatted to a particular precision
<a href="#">nvram_backup</a>	<a href="#">string</a> filename	<a href="#">int</a> bytes written or 0=error	Backup the contents of <a href="#">NVRAM</a> to a file on the SD Card
<a href="#">nvram_defrag</a>			Defragment <a href="#">NVRAM</a> to optimise it
<a href="#">nvram_dump</a>			Dump the contents of <a href="#">NVRAM</a> to standard output
<a href="#">nvram_free</a>		<a href="#">int</a> bytes	Return the number of bytes available in <a href="#">NVRAM</a>
<a href="#">nvram_get</a>	<a href="#">string</a> key	<a href="#">mixed</a> value	Get a value from <a href="#">NVRAM</a>
<a href="#">nvram_restore</a>	<a href="#">string</a> filename		Restore the contents of <a href="#">NVRAM</a> from a file
<a href="#">nvram_set</a>	<a href="#">string</a> key, <a href="#">string</a> value	<a href="#">int</a> 1=OK	Set a <a href="#">key</a> and <a href="#">value</a> in <a href="#">NVRAM</a>
<a href="#">nvram_unset</a>	<a href="#">string</a> key	<a href="#">int</a> 1=OK	Clear a <a href="#">key</a> from <a href="#">NVRAM</a>
<a href="#">ord</a>	<a href="#">string</a> character	<a href="#">int</a> ASCII code	Return the ASCII code for a <a href="#">character</a>
<a href="#">ow_first</a>		<a href="#">array</a> or <a href="#">int</a> 0	Initiate a OneWire bus scan and return the address of the first device found
<a href="#">ow_next</a>		<a href="#">array</a> or <a href="#">int</a> 0	Return the address of the next OneWire device found (after an <a href="#">ow_first</a> )
<a href="#">ow_read</a>		<a href="#">int</a> value or 0	Read a byte from the OneWire bus
<a href="#">ow_read_temp</a>	<a href="#">array</a> device_id	<a href="#">float</a> degrees Celsius	Read a temperature from a device on the OneWire bus
<a href="#">ow_reset</a>			Reset the OneWire bus
<a href="#">ow_write</a>	<a href="#">int</a> value		Write a byte to the OneWire bus
<a href="#">pack</a>	<a href="#">string</a> format, <a href="#">mixed</a> value	<a href="#">string</a>	Pack a value into a <a href="#">string</a>

FUNCTION NAME	PARAMETER(S)	RETURN	DESCRIPTION
phpinfo		string	Return information about the system
pin_configure	int pin_index, int pin_type, int counter_type		Configure an I/O pin as a digital input, output, or analog input
pin_get	int pin_index, int pin_type	int value	Return the value of an I/O pin
pin_set	int pin_index, int value		Set a digital output to value 1 or 0
ping	string host	array	Send an ICMP ping and place the result in an array
power	number base, number exp	number base <sup>exp</sup>	Return base raised to the power of exp
print	string data		Print data to the current output stream such as a web page or terminal
print_r	array		Dump the contents of an array to the current output
printf	string format, mixed values ...		Print a formatted string to standard output
process_kill	int pid		Send a kill request to a process
process_list		array	Return an array of the currently running scripts
rand	int min, int max	int	Return a random integer between min and max
reboot			Reboot the processor
register_callback	string callback_type, string filename, string functionname	int 0 or error code	Register a callback function for system events
rename	string source, string destination	int 0 or error code	Rename or move a file or directory from source to destination
reset			Reset the processor
rmdir	string pathname, int delete_contents	int 0=OK	Remove a directory, with optional deletion of contents
send_sms	string phone_number, string message	int result	sends an sms through a cellular dongle
session_destroy			Clear the current session's data
session_is_new		int	Check if a session was just initiated
session_start			Initiate a new session and send the cookie data for it
set_cert_key	string key, string cert		Set a custom certificate encryption key
set_search_path	string pathname		Set the search path for the telnet client
setethpower	int state		Enable or disable the ethernet controller
setpriority	int priority		Set the priority of the current script
settime	int timestamp, int calibration		Set the system time from a Linux Timestamp, with optional calibration
setusbpower	int state		Enable or disable USB power



sha1	string input	string 40 characters	Calculate the SHA1 hash of a <a href="#">string</a>
sin	number radian_angle	float sine	Return sine of a <a href="#">radian_angle</a>
sizeof	array	int number of elements	Return the number of elements in an <a href="#">array</a>
sleep	int ms		Sleep for specified milliseconds
snmp_trap_send	string message	int 0 or 1	Send an SNMP trap message with ASCII content of <a href="#">source</a>
spi_clearcs			Clear the CS output of the SPI bus
spi_read		int byte	Read a byte from the SPI bus
spi_setcs			Set the CS output of the SPI bus
spi_write	int byte		Write a <a href="#">byte</a> to the SPI bus
sprintf	string format, mixed values ...	string formatted	Return a formatted <a href="#">string</a>
sqr	number number	number squared	Return the square of a <a href="#">number</a>
sqrt	number number	number square root	Return the square root of a <a href="#">number</a>
stats		array	Return system statistics
str_replace	string search, string replace, string subject,[int &count]	string result	Return the string with each occurrence of <a href="#">search</a> replaced with <a href="#">replace</a>
strftime	string format, int timestamp	string formatted	Format a Linux <a href="#">Timestamp</a> using a <a href="#">format string</a>
strlen	string input	int length	Return the length of a <a href="#">string</a>
strpos	string haystack, string needle	int position or -1	Return the position of the first occurrence of a <a href="#">needle</a> in a <a href="#">haystack</a>
strrpos	string haystack, string needle	int position or -1	Return the position of the last occurrence of a <a href="#">needle</a> in a <a href="#">haystack</a>
strtolower	string input	string lowercase	Return the lowercase version of a <a href="#">string</a>
strtoupper	string input	string UPPERCASE	Return the UPPERCASE version of a <a href="#">string</a>
strval	mixed value	string	Return the <a href="#">string</a> equivalent of a <a href="#">number</a>
substr	string input, int start, int length	string substring	Return part of a <a href="#">string</a>
tar_finish	int handle	int 1=OK	Add the ending header to a TAR file
tar_put	int handle, string src_pathname, string tar_pathname	int 1=OK	Add a file to an open file in TAR format
time		int seconds	Return the current system timestamp
timefromfat	int filetime	int seconds	Convert a FAT <a href="#">filetime</a> to a Linux Timestamp
trim	string input	string trimmed	Return the trimmed <a href="#">string</a>
ucfirst	string input	string Lowercase	Convert a <a href="#">string</a> to Lowercase except for the first character
unlink	string filename	int 0 or error code	Remove a file (delete it)
untar	string filename, int verbosity	int 1=OK	Expand a TAR file into the current folder, optionally verbose



<code>unpack</code>	<code>string format, string value</code>	<code>number</code>	Unpack a packed string value and return the original data
<code>uptime</code>		<code>int ms</code>	Return the uptime in milliseconds
<code>urldecode</code>	<code>string str</code>	<code>string</code> string to encode	URL-Decode a string
<code>urlencode</code>	<code>string str</code>	<code>string</code> encoded string	URL-Encode a string
<code>wifi_disable</code>			Disable Wifi module
<code>wifi_enable</code>			Enable Wifi module

From:

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

Permanent link:

<https://www.wattmon.com/dokuwiki/uphp/functions?rev=1636716141>

Last update: **2021/11/12 11:22**

