

# adc\_read

WMPRO FW >= 1.0 WMMEGA FW >= 2.0

Read an onboard ADC channel

## Description

```
int adc_read ( int $channel )
```

Return the raw ADC value of a channel

## Parameter

**\$channel**: Value between 0 and 2

Channel 0 is attached to pin A1, which is the voltage divider to measure battery voltage and can measure up to 60V DC.

Channel 1 is connected to pin A2 which can handle up to 330V DC.

Channel 2 is connected to pin A3, which reads 0-5V DC.

## Return Values

**Integer** ADC value:

WMPRO 10-bit, 0-1023

WMMEGA 12-bit, 0-4095

## Examples

### Example #1

Read Channel 0 and use the constants `v_adc_offset` and `v_adc_mul` to obtain a calibrated voltage<sup>1</sup>:

```
<pre><?
    $adc=adc_read(0);
    $volts=floatval($_GLOBALS['v_adc_mul'])*(floatval($adc)+floatval($_GLOBALS['v_adc_offset']));
    print("adc=".$adc."\r\n");
    print("v_adc_offset=".$_GLOBALS['v_adc_offset']."\r\n");
```

```
print("v_adc_mul=".$_GLOBALS['v_adc_mul']."\r\n");
print("volts=".$volts."\r\n");
?></pre>
```

Typical output for the above example (results vary depending on actual voltage and calibration constants):

```
adc=192
v_adc_offset=24.811435
v_adc_mul=0.065411
volts=14.181854
```

## Additional Information

The raw ADC values can be accessed using `adc_read()`, but it is recommended to use the functions `mb_get_val_by_role()` or `mb_get_dev_by_id()` in order to obtain calculated values for A1, A2 and A3 that are adjusted by the calibration values stored in NVRAM.

### Example #2

Use `mb_get_dev_by_id()` to obtain calibrated values for A1, A2 and A3:

```
<pre><?
$dev=mb_get_dev_by_id(0);
$a1=$dev['a1']/10; // scaled by 10
$a2=$dev['a2']/10; // scaled by 10
$a3=$dev['a3']/100; // scaled by 100
print("A1=".$a1." V\r\n");
print("A2=".$a2." V\r\n");
print("A3=".$a3." V\r\n");
?></pre>
```

Typical output for the above example (results vary depending on actual voltage and calibration constants):

```
A1=13.600001 V
A2=2.100000 V
A3=4.980000 V
```

## See Also

[uphp\\_special\\_variables](#) - Special arrays that are populated automatically (including `$_GLOBALS`)

[pin\\_configure\(\)](#) - Configure an I/O pin as a digital input, output, or analog input [for more information on Channel numbers which are also known as Pin Indexes]

[floatval\(\)](#) - Return the **float** value of a **number** or **string**

1)

**Example #1** demonstrates a legacy method to obtain a calibrated voltage, which was the primary method in early [Wattmon](#) releases, and has not yet been removed from the OS. The calibration constants are loaded automatically into `$_GLOBALS` from the SD Card `/config/battery.ini`. These constants can be calculated and set using Control Panel > Voltage Calibration (Calibrate Voltage Sensor). However, that Control Panel interface does not produce or set calibration constants for Channels 1 and 2, and as mentioned, it stores the constants on the SD Card. This “old method” has been superseded by “a new method” that stores the calibration constants for A1, A2 and A3 in NVRAM. The new calibration method is accessed in Control Panel > Devices. There it is possible to calibrate any device on the [Wattmon](#) or attached by ModBus, and the constants are then stored in NVRAM on the device itself, instead of on the SD Card. See [Additional Information](#) for how to obtain calibrated values using the new method.

From:

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

Permanent link:

[http://wattmon.com/dokuwiki/uphp/functions/adc\\_read](http://wattmon.com/dokuwiki/uphp/functions/adc_read)Last update: **2021/09/13 05:57**