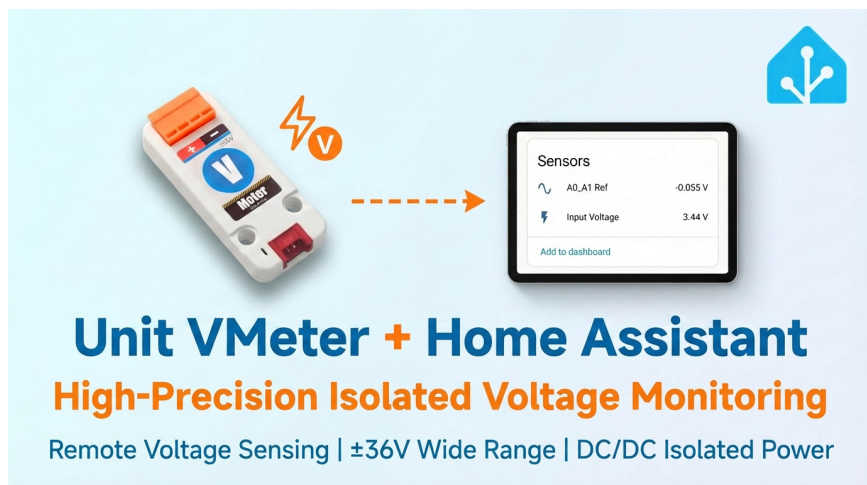


Unit VMeter Home Assistant Integration

This section describes the configuration methods and practical steps for integrating the Unit VMeter voltage sensor into Home Assistant.



Tutorial

- [ADS1115 Datasheet](#)
- Get the latest configuration for ADS1115 from ESPHome: [ADS1115 4-Channel 16-Bit A/D Converter](#)

Note

Unit VMeter is only a standalone sensor platform and requires an additional controller device (such as Atom series, Stamp series, Stick series, Core/Basic series, etc.) to be integrated into Home Assistant.

Learn

1. The [I²C](#) component must be enabled in the ESPHome configuration:

```
# Example configuration entry for ESP32
i2c:
  sda: GPIOXX
  scl: GPIOXX
  scan: true
```

The GPIO pins will vary depending on the controller device used. For example, using Atom Lite as the controller:

```
# I2C Bus on Grove Port (HY2.0-4P)
i2c:
  sda: GPIO26
  scl: GPIO32
```

Unit VMeter Configuration Example:

1. First, configure the basic component ADS1115:

yaml

```
1  ads1115:
2    - address: 0x49
3
4  sensor:
5    - platform: ads1115
6      multiplexer: "A0_A1"
7      gain: 1.024
8      sample_rate: 128
9      name: "Unit VMeter A0_A1 Ref"
10     id: diff
11     update_interval: 10s
```

Note

Unit VMeter only uses two channels, **AIN0** and **AIN1**, so a maximum of 3 **multiplexer** options can be used: "**A0_A1**", "**A0_GND**", "**A1_GND**"; for voltage measurement here, only **A0_A1** is used for voltage calculation.

2. Configure EEPROM to read calibration parameters and combine with a **Template** Sensor to create custom data:

```
1  i2c_device:
2    id: eeprom
3    address: 0x53
4
5  sensor:
6    # Previous ads1115 declarations
7    ...
8  - platform: template
9    name: "Input Voltage"
10   id: input_volt
11   unit_of_measurement: "V"
12   icon: "mdi:flash"
13   accuracy_decimals: 2
14   update_interval: 10s
15   lambda: |-
16
17     float d = id(diff).state;
18
19     if(isnan(d)) return NAN;
20
21     const float PRESSURE_COEFF = 0.015918958f;
22
23     // Remember to change the 'EEPROM_REG' if you changed the 'gain'
24     const uint8_t EEPROM_REG = 0xE8;
25
26     uint8_t calib[8];
27     uint16_t hope, actual;
28
29     // Read EEPROM
30     if ( id(eeprom).read_register(EEPROM_REG, calib, 8) != i2c::ERROR_OK ) {
31       ESP_LOGD("vmeter.sensor", "Failed to read from EEPROM..");
32       return NAN;
33     }
34
35     uint8_t xor_result = 0x00;
36
37     for (uint8_t i = 0; i < 5; i++) {
38       xor_result ^= calib[i];
39     }
40
41     if (xor_result != calib[5]) {
42       return NAN;
43     }
44
45     hope  = (calib[1] << 8) | calib[2];
46     actual = (calib[3] << 8) | calib[4];
47
48     float calibration_ratio = (float)hope / actual;
49
50     ESP_LOGD("vmeter.sensor", "Factory calibration factor: %f", calibration_ratio);
```

```
51
52     float vin = - d / PRESSURE_COEFF;
53     vin = vin * calibration_ratio;
54
55     return vin;
```

3. When reading the EEPROM, the `EEPROM_REG` read address in the lambda expression must be modified based on the `gain` value set for the ADS1115:

C++

```
1 // In lambda expression
2 // remember to change the 'EEPROM_REG' if you changed the 'gain'
3 // for an example, when 'gain' is 1.024
4 const uint8_t EEPROM_REG = 0xE8;
```

ADS1115 Gain	EEPROM Data Register	Max Input Voltage (theory)
6.144	0xD0	-
4.096	0xD8	-
2.048	0xE0	-
1.024	0xE8	64 V
0.512	0xF0	32 V
0.256	0xF8	16 V

The recommended `gain` values are `0.256`, `0.512`, and `1.024`; among them, the resolution is 1mV for measuring voltages below 16V, and 7.9mV for voltages above 16V. Please select the most appropriate `gain` value and read the corresponding calibration parameters in the EEPROM according to the target voltage. For data accuracy, **only one channel (AIN0 or AIN1) can be measured at a time**. Do not connect voltage inputs to both channels simultaneously.

Warning

The optimal measurement range designed for the product is $\pm 36V$. **Do not input voltages outside this range** to avoid device damage. The EEPROM (0x53) contains factory-preset calibration parameters. **Do not perform write operations on the EEPROM**, otherwise, the calibration data will be overwritten, which may lead to inaccurate measurement results.

Quick Start

Once added to the Dashboard, you can view the sensor data in Home Assistant.

 Input Voltage 12.14 V

 Unit VMeter A0_A1 Ref -0.193 V

