QUICK REFERENCE · TK09

Voltmeter

Divides by five, so an analog pin can read up to 25 V.

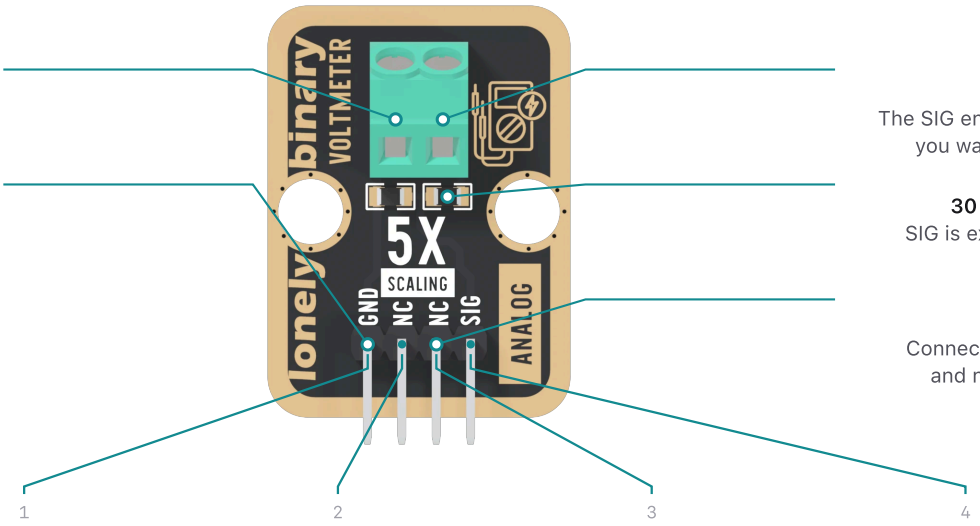
DC only, and never above the limit **At most 25 V on a 5 V board, about 15.5 V on an ESP32.** Positive DC only: a reversed input puts a negative voltage on the pin. Never connect mains, however it is divided.

Pins, and which to wire

Terminal up, header at the bottom. Wire GND and SIG. The two NC pins are connected to nothing, and there is no VCC.

GND screw
The header's GND end. The circuit's ground here.

GND is the square pad
Count from it. The back prints no pin names.



Input screw
The SIG end. The voltage you want to measure.

30 kΩ and 7.5 kΩ
SIG is exactly one fifth of the input.

Both NC pins
Connected to nothing, and no VCC to wire.

Front	GND	NC	NC	SIG
Read it	GND	●	●	SIG

It measures from GND The circuit you measure must share its ground with your board: it cannot read a voltage that floats.

How high each board can go

Arduino Uno

25.0 V at the terminal

SIG to A0. The pin reads to about 5 V.

ESP32, ESP32-S3

15.5 V at the terminal

SIG to GPIO 4. The pin reads to about 3.1 V.

Raspberry Pi Pico

16.5 V at the terminal

SIG to GP26. The pin reads to about 3.3 V.

The input sees 37.5 kΩ: 0.24 mA from a 9 V battery, enough to notice over months, not enough to disturb most circuits. Your analog pin sees 6.0 kΩ.

The ×5 is exact in theory; ±1 % resistors make it good to about 2 %, and one calibration against a multimeter removes most of that. The terminal's own 300 V rating is the terminal's, not the block's.

Read a battery

Arduino IDE · Board: your board · ESP32-S3: USB CDC On Boot: Enabled

```
// SIG's pin. Uno: A0. ESP32: 34. ESP32-S3: 4. Pico: 26.
const int SIG_PIN = A0;
const float SCALE = 5.0;      // the block divides by five
// Uno: 1023 and 5.0. Pico: 1023 and 3.3.
const int ADC_MAX = 1023;
const float VREF = 5.0;

void setup() {
  Serial.begin(115200);
}

void loop() {
  float sig = analogRead(SIG_PIN) * VREF / ADC_MAX;
  // On an ESP32 use analogReadMilliVolts(SIG_PIN) / 1000.0
  Serial.print(sig * SCALE, 2);
  Serial.println(" V");
  delay(500);
}
```

```
9.12 V
9.12 V
9.10 V
```

Serial Monitor at 115200 with a 9 V battery on the terminal, for example.

Wiring

	FRONT	UNO	ESP32	S3	PICO
1	GND	GND	GND	GND	GND
2	NC				
3	NC				
4	SIG	A0	GPIO 34	GPIO 4	GP26

Hole 1 is the square pad. Join the measured circuit's ground to the GND screw. On an ESP32 or ESP32-S3 use an ADC1 pin: ADC2 stops working while Wi-Fi is on.

Calibrate once

Measure one steady voltage with a multimeter and with the sketch. Multiply SCALE by the meter's reading over the sketch's. Averaging a few readings steadies the last digit.

When it does not work

It reads 0 V

The circuit's ground is not joined to the GND screw, or the wires are on the wrong screws: the input goes on the SIG end.

It reads about a fifth of what it should

The sketch forgot to multiply by five.

The ESP32 stops rising

Its ADC reads to about 3.1 V, so about 15.5 V at the terminal. Measure less, or use a 5 V board.

It is a few percent off

Resistor tolerance and the ADC reference. Calibrate once against a multimeter.

It reads something with nothing connected

An open input floats. Short the two screws: it should read 0 V.

It wanders with the battery load

It is measuring truthfully: a battery's voltage sags when it works.

Specifications

Scaling	÷ 5: 30 kΩ over 7.5 kΩ	**Maximum**	25 V on 5 V, about 15.5 V on ESP32
Input resistance	37.5 kΩ	**Input**	Positive DC only, ground shared
Terminal	3.5 mm pitch, 26–16 AWG, strip 4–5 mm	**Header**	4 pins, right-angle, 2.54 mm
Size	22.4 × 30.4 mm, 10.9 mm tall	**Mounting**	2 × Ø 4.8 mm holes, 16 mm apart