

Disc Potentiometer

A thumbwheel on a divider. SIGNAL goes from 0 V to VCC.

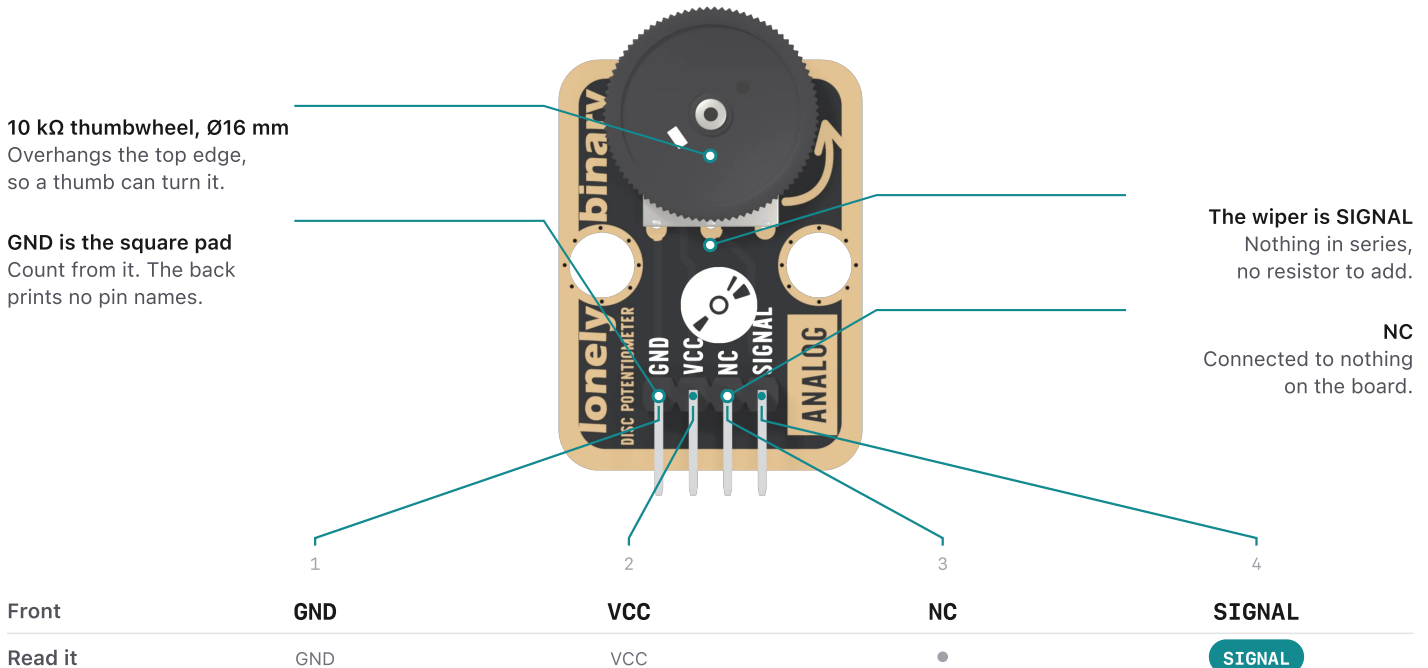


VCC is the top of the range

Turned to the end, SIGNAL is VCC. On an ESP32, ESP32-S3 or Pico, feed VCC from 3V3, never 5V: 5 V would reach a 3.3 V analog pin.

Pins, and which to wire

Wheel up, header at the bottom. Wire GND, VCC and SIGNAL; NC is connected to nothing on the board.



SIGNAL = VCC × position The wheel is a voltage divider. The number you read is a fraction of the ADC's own range.

What each board reads

Arduino Uno

1023 at the far stop

SIGNAL to A0, VCC from 5V. The track draws 0.50 mA.

ESP32-S3

4095 at the far stop

SIGNAL to GPIO 4, VCC from 3V3. The track draws 0.33 mA.

Raspberry Pi Pico

1023 at the far stop

SIGNAL to GP26, VCC from 3V3. The track draws 0.33 mA.

The Uno and Pico measure against the same supply that feeds the wheel, so the reading is a pure ratio. An ESP32 measures against its own reference and flattens near both ends: the last few percent of the turn read the same.

The ESP32 behaves as the ESP32-S3. The track's taper is not stated on the part's drawing; turn it and see which way raises the number.

Read it, as a percentage

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

```
// SIGNAL's pin. Uno: A0. ESP32: 34. ESP32-S3: 4. Pico: 26.
const int POT_PIN = 4;
// Full scale. Uno, Pico: 1023. ESP32, ESP32-S3: 4095.
const int ADC_MAX = 4095;

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

void loop() {
  int raw = analogRead(POT_PIN);
  int pct = (long)raw * 100 / ADC_MAX;
  Serial.print(raw);
  Serial.print("  ");
  Serial.print(pct);
  Serial.println(" %");
  delay(200);
}
```

0 0 %
2047 49 %
4095 100 %

Serial Monitor at 115200, turning the wheel from one stop to the other.

A steadier number

The last digit flickers: that is the ADC's noise, not the wheel. Average eight or sixteen readings, or only act when the value moves by more than a few counts.

When it does not work

It reads 0 whatever I do

VCC is not connected, or SIGNAL is on a different pin from the sketch. Count from the square pad.

The ends read the same for a while

On an ESP32 the ADC flattens near both ends. The Uno and Pico do not.

The number flickers

ADC noise. Average a few readings, or ignore changes of one or two counts.

Specifications

Potentiometer	10 kΩ thumbwheel, single gang	Wheel	Ø16 mm, 2.0 mm thick
Output	0 V to VCC on SIGNAL	Track current	0.50 mA at 5 V, 0.33 mA at 3.3 V
Supply	3.3 V or 5 V, the same as your board	Header	4 pins, right-angle, 2.54 mm
Size	22.4 × 30.4 mm, wheel 3.9 mm past the top	Mounting	2 × Ø 4.8 mm holes, 16 mm apart

Wiring

	FRONT	UNO	ESP32	S3	PICO
1	GND	GND	GND	GND	GND
2	VCC	5V	3V3	3V3	3V3
3	NC				
4	SIGNAL	A0	GPIO 34	GPIO 4	GP26

Hole 1 is the square pad. On an ESP32 or ESP32-S3 use an ADC1 pin: ADC2 stops working while Wi-Fi is on.