

Push Button

Pressed reads HIGH. Released reads LOW.

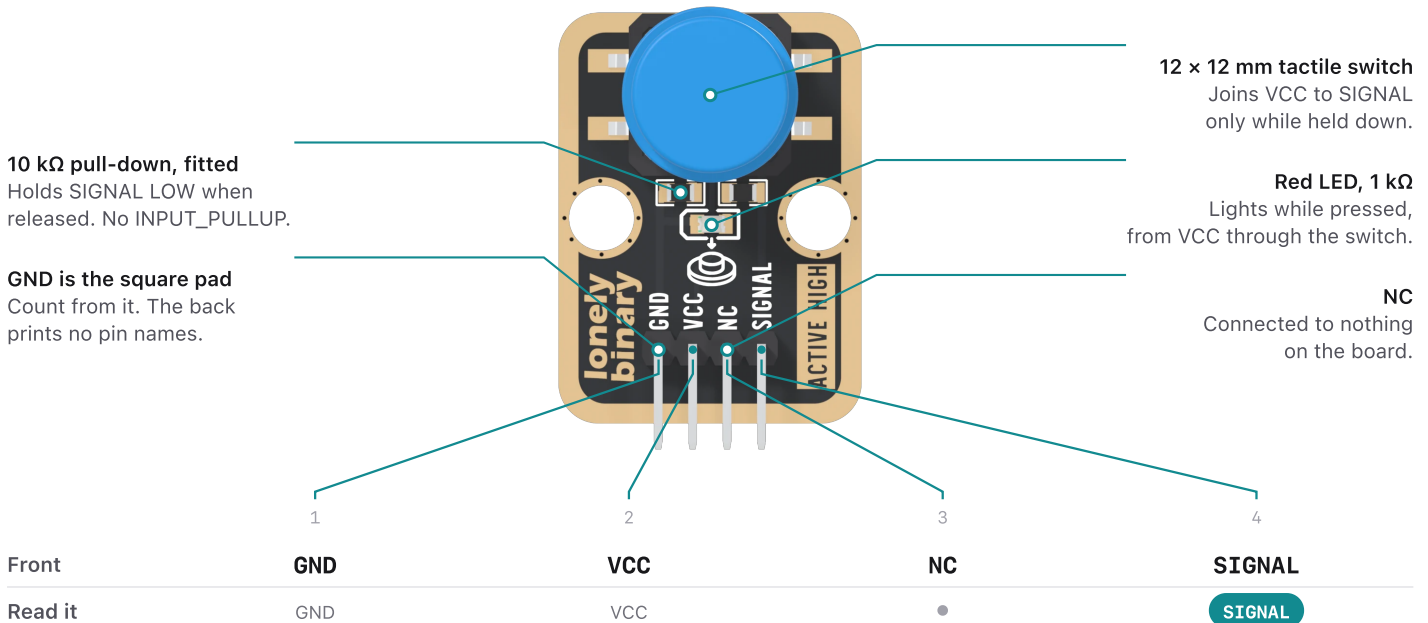


VCC sets the signal

A press connects SIGNAL straight to VCC. On an ESP32, ESP32-S3 or Pico, feed VCC from 3V3, never 5V: 5 V would reach a 3.3 V input on every press.

Four pins, three wires

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



Pressed = HIGH

This board has a pull-down, not a pull-up. Use INPUT, and treat HIGH as the press.

VCC is the signal

Arduino Uno

5V to VCC

A press reads 5 V. The LED takes about 3.0 mA; the whole press about 3.5 mA.

ESP32, ESP32-S3, Pico

3V3 to VCC

A press reads 3.3 V. The LED takes about 1.3 mA, a little dimmer; the whole press about 1.6 mA.

Pressed, SIGNAL is joined to VCC, so whatever you feed VCC is the voltage your input sees. Released, the pull-down holds it at 0 V and the block draws nothing. With VCC unconnected a press reads LOW like everything else.

LED currents are worked out from a red LED's typical drop (about 2.0 V across it), not measured. The pull-down adds 10 kΩ's share while pressed.

Count presses, once each

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

```
// SIGNAL's pin. Uno: 2. ESP32: 25. ESP32-S3: 4. Pico: 15.
const int BUTTON_PIN = 4;
const unsigned long SETTLE_MS = 20; // ignore bounce
bool stable = LOW, lastRaw = LOW;
unsigned long lastChange = 0;
unsigned long presses = 0;

void setup() {
  Serial.begin(115200);
  pinMode(BUTTON_PIN, INPUT); // the board has its own pull-down
}

void loop() {
  bool raw = digitalRead(BUTTON_PIN);
  if (raw != lastRaw) { // it moved: start the settle timer
    lastChange = millis();
    lastRaw = raw;
  }
  if (millis() - lastChange >= SETTLE_MS && raw != stable) {
    stable = raw;
    if (stable == HIGH) { // pressed reads HIGH on this board
      presses++;
      Serial.println(presses);
    }
  }
}
```

1
2
3

Serial Monitor at 115200: one number per press, however fast.

Why one press counts as several

A mechanical contact bounces for a few milliseconds as it closes and opens, and a loop reading thousands of times a second sees each bounce as a press. The sketch only accepts a reading that has held still for 20 ms. If a press still counts twice, try 30 to 50.

When it does not work

A press does nothing

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

One press counts two or three times

Contact bounce. Accept a change only after it has held still for 20 ms, as the sketch does.

It reads backwards

A sketch written for a pull-up button treats LOW as pressed. On this board pressed reads HIGH.

Specifications

Switch	12 × 12 mm tactile, 0.3 mm travel	Switch rating	12 V DC, 50 mA
Pull-down	10 kΩ, SIGNAL to GND	Indicator	Red LED, 1 kΩ, lit while pressed
Logic	Active high: pressed reads HIGH	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, 2 × Ø 4.8 mm holes

Wiring

	FRONT	UNO	ESP32	S3	PICO
1	GND	GND	GND	GND	GND
2	VCC	5V	3V3	3V3	3V3
3	NC				
4	SIGNAL	D2	GPIO 25	GPIO 4	GP15

Hole 1 is the square pad. VCC is 3V3 on every 3.3 V board: 5 V there would put 5 V on a 3.3 V input with every press.