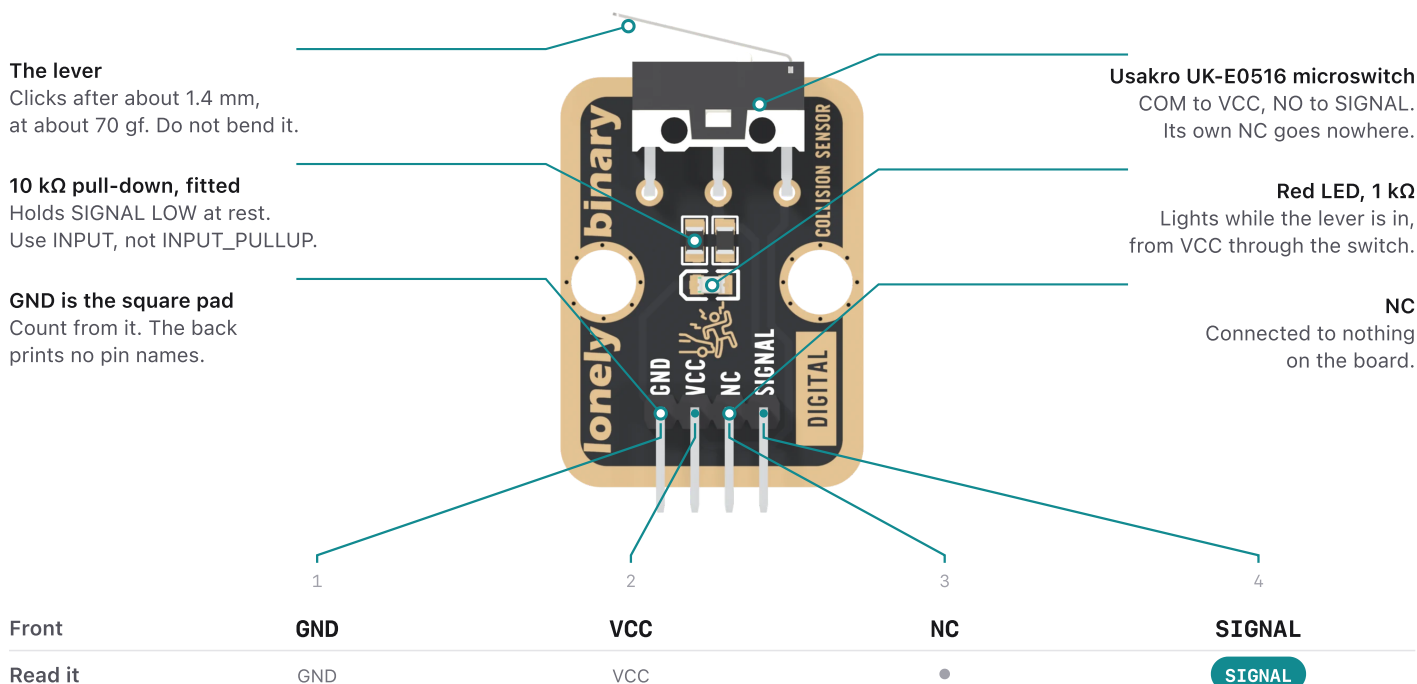




A hit connects SIGNAL straight to VCC. **On an ESP32, ESP32-S3 or Pico, feed VCC from 3V3, never 5V**, and never from a motor battery: whatever is on VCC reaches your input on every hit.

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



Hit = HIGH This board has a pull-down, not a pull-up, and the switch's NO contact on SIGNAL. Use INPUT, and treat HIGH as a hit.

VCC is the signal

5V to VCC

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

3V3 to VCC

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

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

LED currents are worked out from a red LED's typical 2.0 V drop, not measured. Lever travel is the switch drawing's free position minus its operating position, 10.2 and 8.8 mm, each ± 0.8 mm.

Count hits, 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 BUMPER_PIN = 4;
const unsigned long DEBOUNCE_MS = 20; // ignore bounce
int lastReading = LOW, state = LOW;
unsigned long lastChange = 0, hits = 0;

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

void loop() {
  int reading = digitalRead(BUMPER_PIN);
  if (reading != lastReading) { // it moved: restart the clock
    lastReading = reading;
    lastChange = millis();
  }
  if (millis() - lastChange >= DEBOUNCE_MS && reading != state) {
    state = reading;
    if (state == HIGH) { // a hit reads HIGH on this board
      hits++;
      Serial.println(hits);
    }
  }
}
```

1
2
3
Serial Monitor at 115200: one number per hit, however hard.

On a robot, back off

Mount the block with its top edge forward, so the lever meets an obstacle first. On a hit, reverse for about 400 ms, turn for about 300 ms, and drive on, timing each step with millis() rather than delay() so the bumper is still read while the robot turns. The 20 ms window keeps a glancing hit; do not raise it far.

When it does not work

A hit does nothing, LED dark

VCC or GND is not connected, or the cable is one pin over. Count from the square pad: GND, VCC, NC, SIGNAL.

It reads a hit all the time

A sketch testing for LOW, or INPUT_PULLUP. If the LED is lit too, something on the robot is resting on the lever.

It misses glancing hits

Only the lever triggers it, after about 1.4 mm. Make it the part that sticks out furthest, and keep the window short.

Specifications

Switch	Usakro UK-E0516 lever microswitch	Lever	Clicks after about 1.4 mm, about 70 gf
Switch rating	48 V DC, 0.1 A, 1 000 000 operations	Pull-down	10 kΩ, SIGNAL to GND
Logic	Active high: a hit 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, 7.3 mm tall

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 hit. These are the TK04 push button's pins too.