QUICK REFERENCE · TK05

Latching Button

Press on, press again off. On reads HIGH.

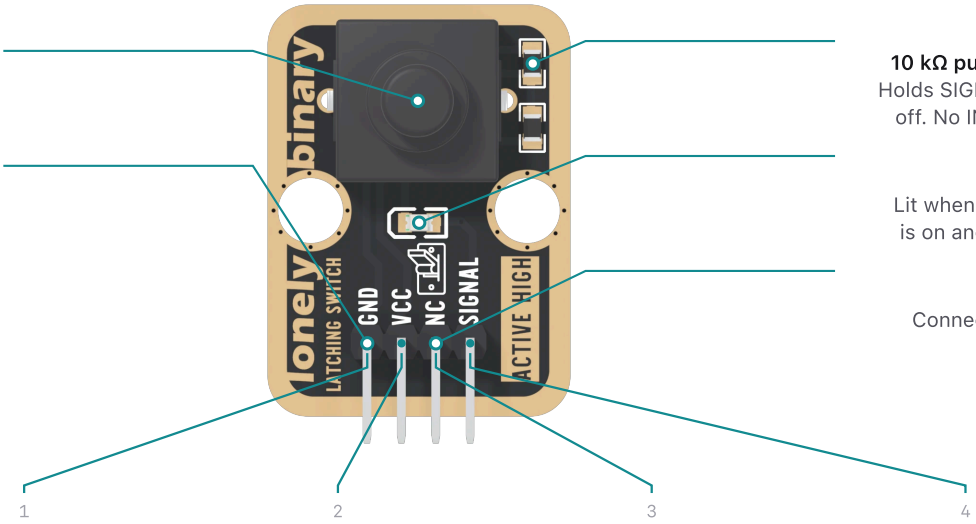
VCC sets the signal While on, SIGNAL is joined straight to VCC. **On an ESP32, ESP32-S3 or Pico, feed VCC from 3V3, never 5V**: 5 V would sit on a 3.3 V input for as long as the switch is on.

Four pins, three wires

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

Latching switch
Press: on. Press again: off. 700 gf, a firm press.

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



10 kΩ pull-down, fitted
Holds SIGNAL LOW while off. No INPUT_PULLUP.

Red LED, 1 kΩ
Lit whenever the switch is on and VCC is wired.

NC
Connected to nothing on the board.

Front	GND	VCC	NC	SIGNAL
Read it	GND	VCC	●	SIGNAL

On = HIGH This board has a pull-down, not a pull-up. Use INPUT, and read HIGH as on.

VCC is the signal

Arduino Uno

5V to VCC

On reads 5 V. The LED takes about 3.0 mA; the block about 3.5 mA while on, nothing while off.

ESP32, ESP32-S3, Pico

3V3 to VCC

On reads 3.3 V. The LED takes about 1.3 mA, a little dimmer; the block about 1.6 mA while on.

On, SIGNAL is joined to VCC, so whatever you feed VCC is the voltage your input sees. Off, the pull-down holds it at 0 V. The switch stays where it was left through a reset or a power cut, so the sketch should read it in setup().

LED currents are worked out from a typical red LED (about 2.0 V across it), not measured. Switch ratings are from the switch's drawing.

Read it, then watch it

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 SWITCH_PIN = 4;
const unsigned long SETTLE_MS = 20;   // ignore bounce

bool stable, lastRaw;
unsigned long lastChange = 0;

void setup() {
  Serial.begin(115200);
  pinMode(SWITCH_PIN, INPUT);   // the board has its own pull-down
  stable = lastRaw = digitalRead(SWITCH_PIN);
  // The switch kept its position through the reset: read it first.
  Serial.println(stable == HIGH ? "started ON" : "started OFF");
}

void loop() {
  bool raw = digitalRead(SWITCH_PIN);
  if (raw != lastRaw) {          // it moved: start the settle timer
    lastChange = millis();
    lastRaw = raw;
  }
  if (millis() - lastChange >= SETTLE_MS && raw != stable) {
    stable = raw;                // a real change, reported once
    Serial.println(stable == HIGH ? "turned ON" : "turned OFF");
  }
}
```

```
started OFF
turned ON
turned OFF
```

Serial Monitor at 115200: the position at start, then one line per change.

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. The same pins as the TK04 push button, so the two blocks swap without rewiring.

A level, not an event

digitalRead tells you where the switch is, not that it was pressed. To act once per change, remember the last settled reading and compare. The contact still bounces for a few milliseconds at each change, so accept a changed level only after 20 ms.

When it does not work

It reads LOW whatever I do

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

It reads HIGH all the time

The switch is latched on: press it once. If the LED is dark as well, SIGNAL is wired to a supply pin.

One press prints two changes

Contact bounce at the change. Accept a changed level only after it has held for 20 ms.

The sketch starts in the wrong mode

Read the switch in setup(). It kept its position through the reset.

It reads backwards

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

The LED is dimmer on an ESP32

Normal: 3.3 V leaves less above the red LED's 2.0 V. The reading is unaffected.

Specifications

Switch	12 × 12 mm, push on, push off	**Switch rating**	30 V DC, 1 A, 50,000 cycles
Force	700 ± 100 gf	**Pull-down**	10 kΩ, SIGNAL to GND
Indicator	Red LED, 1 kΩ, lit while on	**Logic**	Active high: on reads HIGH
Header	4 pins, right-angle, 2.54 mm	**Size**	22.4 × 30.4 mm, 9 mm to the button top