

Dual Axis Joystick

A thumbstick with a centre click: two analog axes and one digital pin.



VCC is the ceiling

X and Y reach VCC at the ends of the travel and SW is joined to VCC on every click. On an ESP32, ESP32-S3 or Pico, wire VCC to 3V3, never 5V. On an Uno, 5V.

Pins, and which to wire

Parts up, header at the bottom. Wire GND, VCC, X, Y and SW; NC is connected to nothing on the board.

Pull-down
Holds SW LOW until the stick is pressed.

SW light, blue
On while pressed. Dim on 3.3 V, not a fault.

PWR light, red
On whenever VCC and GND are right.

GND is the square pad
Count from it. The back prints the names mirrored.

4.8 mm holes
24 mm apart; the board is 31.0 mm square.

Thumbstick
Two pots, X and Y, and a click under the cap.

Front	GND	VCC	NC	X	Y	SW
Back	GND	VCC	NC	X	Y	SW
Read it	GND	VCC	•	X	Y	SW

SW is HIGH when pressed

A resistor on the board pulls SW down and the click joins it to VCC: LOW at rest, HIGH pressed. Set the pin to INPUT. INPUT_PULLUP only fights the board.

What it reads

Stick let go

$\frac{1}{2}$ of VCC, near

About 512 of 1023 on an Uno or a Pico, 2048 of 4095 on an ESP32. Never exactly: measure it at start-up.

Stick at an end

0 / VCC on X or Y

0 or 1023 on an Uno with VCC on 5V. With VCC on 3V3 an Uno only reaches 675.

Stick pressed

HIGH on SW

LOW at rest. The blue SW light comes on while you hold it.

Treat X and Y as distances from a centre you measured, not from half of full scale, and call anything within 10 per cent of it zero. X and Y are the pots' names, not directions: push the stick each way once and see which moves.

Rest points vary from stick to stick and are not measured. The ESP32 reads nothing below about 0.1 V and tops out near 3.1 V, so its ends read flat. Worked out from the net list, not measured on a board.

A centre and a dead zone

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

```
// X Y SW. Uno A0 A1 2. ESP32 34 35 25. S3 4 5 7. Pico 26 27 15.
const int X_PIN = 4, Y_PIN = 5, SW_PIN = 7;
const int TOP = 4095; // ESP32s 4095; Uno, Pico 1023
const int DEAD_ZONE = 10; // per cent of each half
int cx, cy;

int pct(int v, int centre) { // -100..100 around the centre
  int span = v >= centre ? TOP - centre : centre;
  int p = (long)(v - centre) * 100 / span;
  return abs(p) < DEAD_ZONE ? 0 : p;
}

void setup() {
  Serial.begin(115200);
  pinMode(SW_PIN, INPUT); // pulled down on the board
  delay(200); // hands off the stick
  cx = analogRead(X_PIN);
  cy = analogRead(Y_PIN);
}

void loop() {
  Serial.print(pct(analogRead(X_PIN), cx)); Serial.print(" ");
  Serial.print(pct(analogRead(Y_PIN), cy)); Serial.print(" ");
  Serial.println(digitalRead(SW_PIN));
  delay(100);
}
```

0 0 0
100 0 0
0 -64 0
0 0 1

Serial Monitor at 115200: X and Y in per cent, then SW.

The click bounces

The switch's contacts chatter for a few milliseconds as they close and open, so a fast loop can count one press several times. Count the change from LOW to HIGH, then leave SW alone for 20 ms. While pressed, the 5.1 kΩ pull-down takes about 0.98 mA from a 5 V supply.

When it does not work

PWR light dark

The red one. VCC or GND is missing, or the two are swapped. Count from the square pad.

SW never HIGH

If the blue SW light comes on, the wire or pin number is wrong. Test for HIGH, not LOW.

Creeps when let go

Measure the centre at start-up with the stick let go, and add a dead zone.

Specifications

Stick	Two pots, X and Y; at least 60° end to end	Click	SW, 5.1 kΩ pull-down, HIGH when pressed
Lights	PWR red; SW blue, while pressed	Supply	3.3 V or 5 V, the same as your board
Pins	5 to wire; NC in no net	Header	6 right-angle male pins, 2.54 mm
Size	31.0 × 31.0 mm, 4 mounting holes	Height	About 29 mm to the top of the cap

Wiring

	FRONT	UNO	ESP32	S3	PICO
1	GND	GND	GND	GND	GND
2	VCC	5V	3V3	3V3	3V3
3	NC				
4	X	A0	GPIO 34	GPIO 4	GP26
5	Y	A1	GPIO 35	GPIO 5	GP27
6	SW	D2	GPIO 25	GPIO 7	GP15

Hole 1, at the left, is the square pad. X and Y need analog pins; on the ESP32s, ADC1 pins keep working with Wi-Fi on. Leave NC empty.