

Slim Joystick

A thumb stick, a centre click and six buttons on four signal pins.



3V3, on every board

The header prints **3V3**. Wire it to 3V3 on an ESP32, ESP32-S3, Pico and Uno. Every signal is a share of it, so the button levels below only hold at 3.3 V.

Pins, and which to wire

Parts up, header along the top. All six pins are wired: the square pad is GND, at the right-hand end.

10 kΩ ladder, ×6

Gives each button its own share of 3V3.

Thumb stick, X and Y

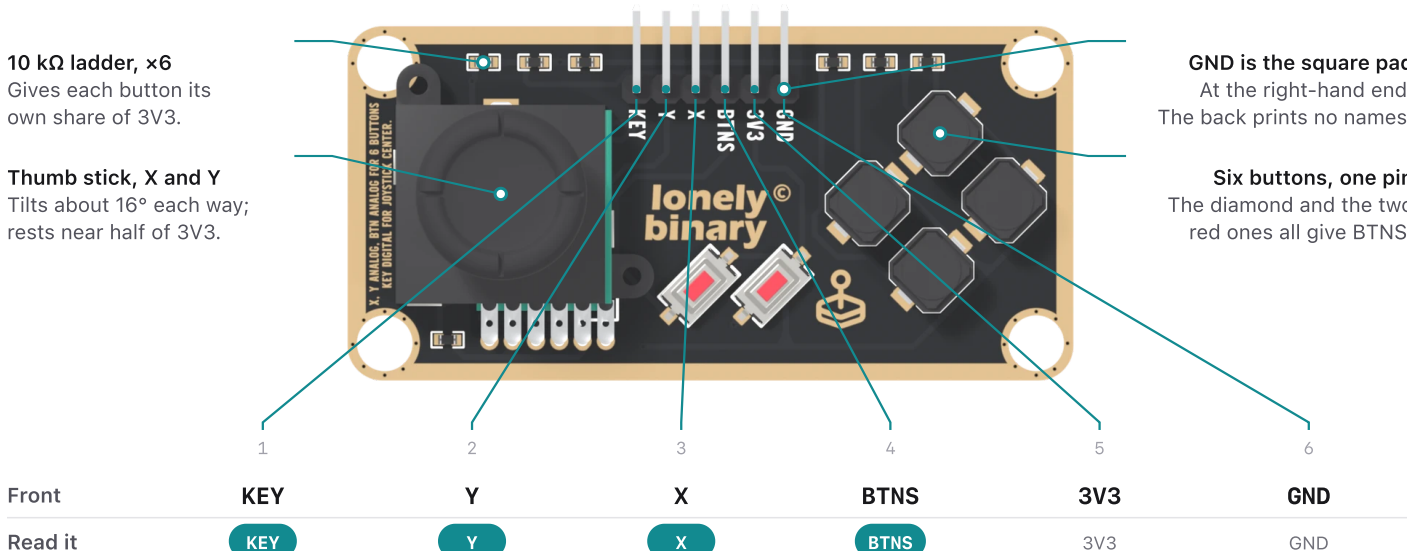
Tilts about 16° each way; rests near half of 3V3.

GND is the square pad

At the right-hand end. The back prints no names.

Six buttons, one pin

The diamond and the two red ones all give BTNS.



BTNS is analog

Six buttons share one pin. Each gives its own voltage; nothing pressed is 0 V. Read BTNS with `analogRead` and pick the nearest level. Two held at once read as one of them.

Which button, which level

U11, red, left

all of 3V3

3300 mV. About 675 on an Uno.

U10, red, right

1/2 of 3V3

1650 mV. About 338 on an Uno.

U5, diamond, bottom

1/3 of 3V3

1100 mV. About 225 on an Uno.

U7, diamond, left

1/4 of 3V3

825 mV. About 169 on an Uno.

U8, diamond, top

1/5 of 3V3

660 mV. About 135 on an Uno.

U6, diamond, right

1/6 of 3V3

550 mV. About 113 on an Uno.

Pick the button by the band BTNS falls in: halfway between neighbouring levels. With two held, the one higher in this list wins outright, so a diagonal on the diamond reads as one button.

Levels from the net list: 10 kΩ steps, $VCC / (1 + k)$. The Uno measures against 5 V, so with 3V3 on the header its readings top out near 675. Worked out, not measured.

Read all eight

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

```
// X Y BTNS KEY. Uno A0 A1 A2 2. ESP32 34 35 32 25.
// S3 4 5 6 7. Pico 26 27 28 15.
const int X_PIN = 4, Y_PIN = 5, BTNS_PIN = 6, KEY_PIN = 7;
const float FULL_MV = 5000.0; // Uno 5000, Pico 3300
const char* NAME[] = {"U11", "U10", "U5", "U7", "U8", "U6"};
const int LIMIT_MV[] = {2475, 1375, 963, 743, 605, 275};

float readMv(int pin) {
  #if defined(ARDUINO_ARCH_ESP32)
    return analogReadMilliVolts(pin);
  #else
    return analogRead(pin) * FULL_MV / 1023.0;
  #endif
}

void setup() {
  Serial.begin(115200);
  pinMode(KEY_PIN, INPUT); // pulled down on the board
}

void loop() {
  float mv = readMv(BTNS_PIN);
  int b = -1;
  for (int i = 0; i < 6 && b < 0; i++) if (mv > LIMIT_MV[i]) b = i;
  Serial.print(analogRead(X_PIN)); Serial.print(" ");
  Serial.print(analogRead(Y_PIN)); Serial.print(" ");
  Serial.print(digitalRead(KEY_PIN)); Serial.print(" ");
  Serial.println(b < 0 ? "none" : NAME[b]);
  delay(200);
}
```

338 345 0 none
338 345 0 U7
12 345 1 none

Serial Monitor at 115200, on an Uno: X, Y, KEY and the button held.

The stick and its click

X and Y rest near the middle but rarely exactly on it. Read both at start-up with the stick untouched, call that the centre, and ignore a little either side. Which way each axis rises depends on how you hold the board. KEY is a contact: it bounces, so wait 20 ms before trusting a change. The ladder passes at most 0.33 mA.

When it does not work

A button reads as another

3V3 is on 5V, or the limits are for another board. Wire 3V3 and use millivolts.

KEY never reads HIGH

The click joins KEY to 3V3: check the 3V3 wire and the KEY pin. Use INPUT; the board pulls it down.

The stick drifts at rest

The centre is not exactly half. Measure it at start-up and add a dead zone.

Specifications

Stick	Two pots, X and Y; about 16° each way	Click	KEY, 10 kΩ pull-down, HIGH when pushed
Buttons	6 on BTNS: 2 red, 4 in a diamond	Levels	all, 1/2, 1/3, 1/4, 1/5, 1/6 of 3V3
Supply	3V3 on every board, as printed	Header	6 pins along the top, 2.54 mm
Size	62.4 × 30.4 mm, 4 mounting holes	Height	about 13.5 mm to the top of the stick

Wiring

	FRONT	UNO	ESP32	S3	PICO
1	KEY	D2	GPIO 25	GPIO 7	GP15
2	Y	A1	GPIO 35	GPIO 5	GP27
3	X	A0	GPIO 34	GPIO 4	GP26
4	BTNS	A2	GPIO 32	GPIO 6	GP28
5	3V3	3V3	3V3	3V3	3V3
6	GND	GND	GND	GND	GND

Hole 6, at the right, is the square pad. X, Y and BTNS need analog pins; on the ESP32s, ADC1 pins keep working with Wi-Fi on.