

USB-C Breakout

A USB-C socket with both 5.1 k Ω CC resistors fitted, so a charger switches 5 V on.



VBUS is 5 V: never wire it to a 3V3 pin

A 3V3 pin is a regulator's output. 5 V fed backwards into it can destroy the regulator and every part on the 3.3 V rail. **VBUS** goes to a 5V or VIN pin, or to your circuit's 5 V rail.

Pins, and which to wire

Drawn with the socket at the top, so the six holes run along the bottom: GND, the square pad, is at the right-hand end. Both faces print the same six names.

USB-C socket

Rated 5 V 3 A. USB 2.0 pair only, no SuperSpeed.

R1, R2: 5.1 k Ω to GND

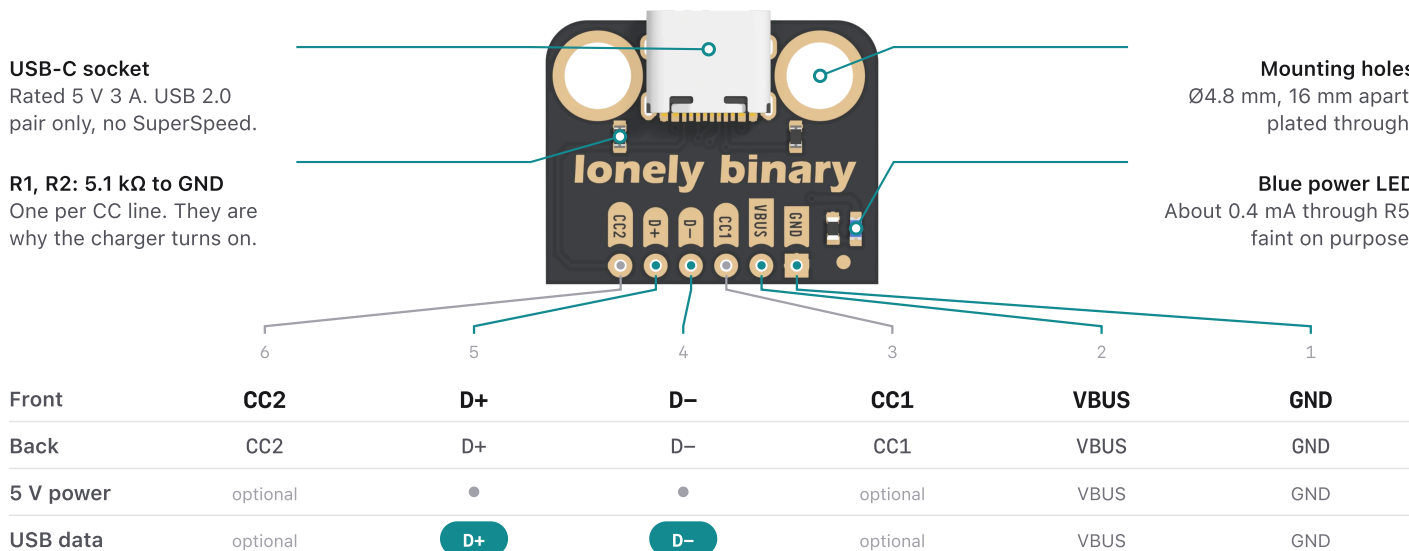
One per CC line. They are why the charger turns on.

Mounting holes

Ø4.8 mm, 16 mm apart, plated through.

Blue power LED

About 0.4 mA through R5, faint on purpose.



A socket is not a supply

A USB-C charger keeps VBUS at 0 V until it sees 5.1 k Ω to ground on a CC line. A bare socket reads 0 V for that reason, and so does this board on a USB-A to USB-C cable, which has no CC wire. Use a C-to-C cable.

What the charger offers, read on CC

Default USB

0.42 V on CC

500 mA. What every source offers, and all you may count on without measuring.

1.5 A source

0.94 V on CC

Most phone chargers. The charger's own 22 k Ω pull-up against this board's 5.1 k Ω .

3 A source

1.69 V on CC

The most a 5 V USB-C source offers, and the socket's own 3 A rating.

Only one CC line reads a voltage; the other is the one the cable left open, and turning the plug over swaps them. Nothing on the board reads CC, so without measuring, plan for 500 mA.

The voltages are worked out from the USB Type-C pull-up values and 5.1 k Ω , not measured on this board; real resistor tolerances move them by about a tenth of a volt. The socket rating is SHOU HAN's specification for the TYPE-C 16PIN 2MD(073).

Read what the charger offers

Arduino IDE · Board: ESP32 Dev Module · No libraries

```
// ESP32 Dev Module. Wiring: table on the right.
const int CC1_PIN = 34, CC2_PIN = 35;
const float EDGE_1V5 = 0.66, EDGE_3A = 1.23;
const float OPEN_MAX = 0.20; // below this: open line

float readVolts(int pin) {
  uint32_t mv = 0;
  for (int i = 0; i < 16; i++) mv += analogReadMilliVolts(pin);
  return mv / 16.0 / 1000.0;
}

const char *advertises(float v) {
  if (v < EDGE_1V5) return "500 mA (default USB)";
  if (v < EDGE_3A)  return "1.5 A";
  return "3.0 A";
}

void setup() {
  Serial.begin(115200);
}

void loop() {
  float v1 = readVolts(CC1_PIN), v2 = readVolts(CC2_PIN);
  bool onCC1 = v1 > v2;
  float live = onCC1 ? v1 : v2;
  Serial.printf("CC1 %.2f V   CC2 %.2f V\n", v1, v2);
  if (live < OPEN_MAX)
    Serial.println("  both open: no source, or no CC wire");
  else
    Serial.printf("  cable on %s, source offers %s\n",
                  onCC1 ? "CC1" : "CC2", advertises(live));
  delay(1000);
}
```

CC1 0.94 V CC2 0.00 V
cable on CC1, source offers 1.5 A
Serial Monitor at 115200, once a second. Turn the plug over and CC1 and CC2 swap.

Taking power, never giving it

The 5.1 kΩ pull-downs make the board a sink: it takes power from a charger. A socket that hands power out, to charge a phone, needs pull-ups to 5 V on CC instead, so two of these boards face to face, or this board fed from your own 5 V supply, detect nothing and nothing flows. It does not negotiate either: 9 V and 12 V need a Power Delivery chip, which this board does not have.

When it does not work

VBUS is 0 V and the LED is dark

No handshake. Try a C-to-C cable, then another charger. A USB-A to C cable has no CC wire.

Works on one charger, not another

Strict chargers check CC; lenient ones do not. Believe the strict one and check the wiring.

D+ and D− wired, nothing enumerates

D− is hole 4, D+ hole 5. Check the pair is not crossed and GND is shared.

Specifications

Socket	USB-C, 16 contacts, rated 5 V 3 A
Negotiates	Nothing; 5 V only, no PD
Direction	Sink only; not a host port
Durability	10000 insertions

Wiring

	FRONT	CC WATCH	POWER
1	GND	GND	GND
2	VBUS	—	5V/VIN
3	CC1	34	—
4	D−	—	—
5	D+	—	—
6	CC2	35	—

CC watch: power the ESP32 from its own USB cable and leave VBUS unwired, so two supplies are never joined. Power: VBUS to a 5V or VIN pin, never 3V3. CC never rises above about 1.7 V, so it is safe on an ESP32 input.

VBUS sags under load

More than the source offers, or thin jumpers. Measure at the board and at the circuit.

I wired VBUS to a 3V3 pin

Move it to 5V or VIN. A board with neither wants 3.3 V from a regulator.

One CC reads about 0.4 V, the other 0 V

Correct. The cable connects one CC line; turn the plug over and they swap.