

ESP32-C3 OLED Screw Terminal Base

Two blocks of eight screw terminals for the sixteen holes, a battery socket, a charger and a power switch.

✗ Seat the board the right way round, cable out

The board fits end for end as well, and that way its 5V, ground and 3V3 holes land on the base's 3, 4 and 5 terminals: the battery rail drives three chip pins. Put V5 on the board and 5V on the base at the same end, with the switch off and the USB cable out.

Left block, top to bottom

Slide switch

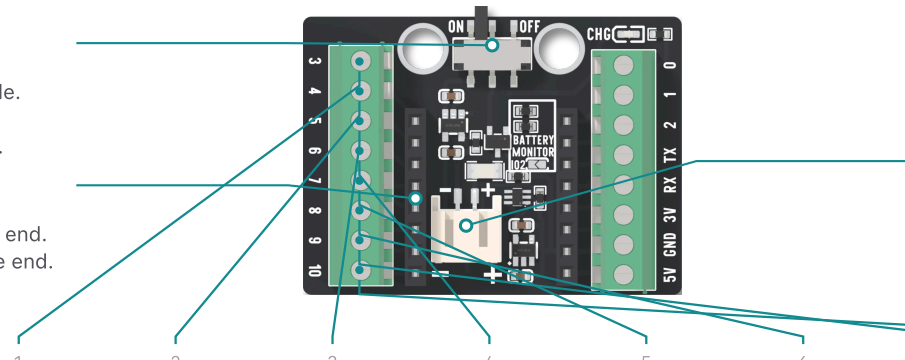
ON and OFF either side.
OFF takes the cell out of the charger as well.

Socket for the board

Antenna at the switch end.
Match V5 to 5V at one end.

Battery socket

JST PH 2.0. The + on the silkscreen is the positive pin. Check the lead.



	1	2	3	4	5	6	7	8	9	10
Front	3	4	5	6	7	8	9	10		
Digital, analog	GPIO 3	GPIO 4			GPIO 7					GPIO 10
I2C with display			SDA	SCL						
Serial, UART0										

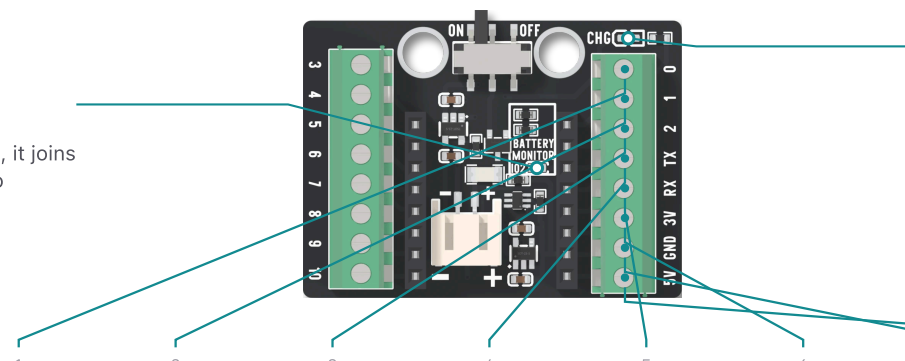
Right block, top to bottom

BATTERY MONITOR

Open jumper. Bridged, it joins half the cell voltage to terminal 2 for good.

CHG LED

Steady while the cell charges, switch ON.



	1	2	3	4	5	6	7	8
Front	0	1	2	TX	RX	3V	GND	5V
Digital, analog	GPIO 0	GPIO 1	GPIO 2			3V	GND	5V
I2C with display						3V	GND	5V
Serial, UART0				TX, GPIO 21	RX, GPIO 20	3V	GND	5V

The left block has no power

3V, GND and 5V are one terminal each, all on the right block. A module wired to the left block takes its supply and its ground from there. Terminals 5 and 6 are the display's bus; 8 is the blue LED, 9 the BOO button, TX and RX the serial port. A cell needs its own protection circuit, and charging needs the switch ON.

The first program, a potentiometer on GPIO 1

Arduino IDE · Board ESP32C3 Dev Module · USB CDC On Boot Enabled · Flash 4MB · one potentiometer, three wires

```
// ESP32-C3 OLED screw terminal base: read a potentiometer on GPIO 1.
//
// Wiring, all on the right-hand block:
//   pot end A -> 3V       pot wiper -> terminal 1       pot end B -> GND
//
// Arduino IDE, Tools menu:
//   Board           ESP32C3 Dev Module
//   USB CDC On Boot Enabled
//   Flash Size      4MB (32Mb)
// No library needed.

const int POT = 1;

void setup() {
  Serial.begin(115200);
  delay(1500); // let the USB port enrol
}

void loop() {
  Serial.printf("raw %4d\n", analogRead(POT));
  delay(500);
}
```

raw 2048

The number runs from near 0 to near 4095 as the knob turns. A reading that wanders without following the knob means the wiper wire is not under its terminal.

The three wires

TERMINAL	WHERE	WHAT	GOES TO
3V	right, 6th down	supply	one end of the track
1	right, 2nd down	GPIO 1, ADC	the wiper
GND	right, 7th down	ground	the other end of the track

When it does not work

A module on the left block has no power

The left block is eight signals and nothing else: GPIO 3 to 10. 3V, GND and 5V are on the right block, one terminal each. A module wired to the left block needs its supply and its ground wired across from the right block, or taken from a breadboard that both blocks feed.

A sensor on terminal 5 or 6 makes the screen misbehave

Those two terminals are the display's bus: GPIO 5 is SDA and GPIO 6 is SCL, with the screen already on them. Another I2C device with a different address belongs there. An LED, a button or a sensor's data line does not, and it will corrupt what the screen is sent.

The reading never moves

The wiper wire is in the wrong terminal, or in the right one with the screw closed on its insulation. Check that the wire goes under the terminal printed 1, that bare copper is in the clamp, and that it does not move when tugged. A pin with nothing on it floats and drifts, so a reading that wanders but does not follow the knob means the wiper is not connected.

A wire pulls out even though the screw is tight

The clamp has closed on insulation, or beside the copper. Back the screw out until you can see the clamp open, put bare copper under it with no insulation inside the clamp, tighten until the wire will not move, and tug it. A screw that has closed on insulation feels tight and tests dead.

GPIO 2 reads a voltage nobody applied

The BATTERY MONITOR jumper has been bridged, and the terminal printed 2 is joined to the middle of the battery divider. It reads half the battery voltage, whatever you wired to it. The jumper ships open; cut or scrape the solder bridge to free the pin again.

The Serial Monitor shows nothing

USB CDC On Boot is set to Disabled, which is the usual cause on this board. Set it to Enabled, upload again, and reopen the monitor at 115200. The sketch runs and prints into nothing otherwise.

Specifications

Fits	ESP32-C3 OLED board, rows 17.78 mm apart	Terminals	2 × 8-way, 3.5 mm pitch, screw
GPIOs at the edge	13, each once	Power ways	3V, GND, 5V, right block only
Battery	JST PH 2.0, one 3.7 V cell with protection	Charger	TP4054, to 4.2 V, switch ON
Mounting	2 holes, 4.8 mm, 16 mm apart	Size	44.49 × 32.54 mm, 8.7 mm tall