

ESP32 Gold Edition

38 holes, one USB-C port, a DAC the S3 does not have, and six pads that go nowhere.

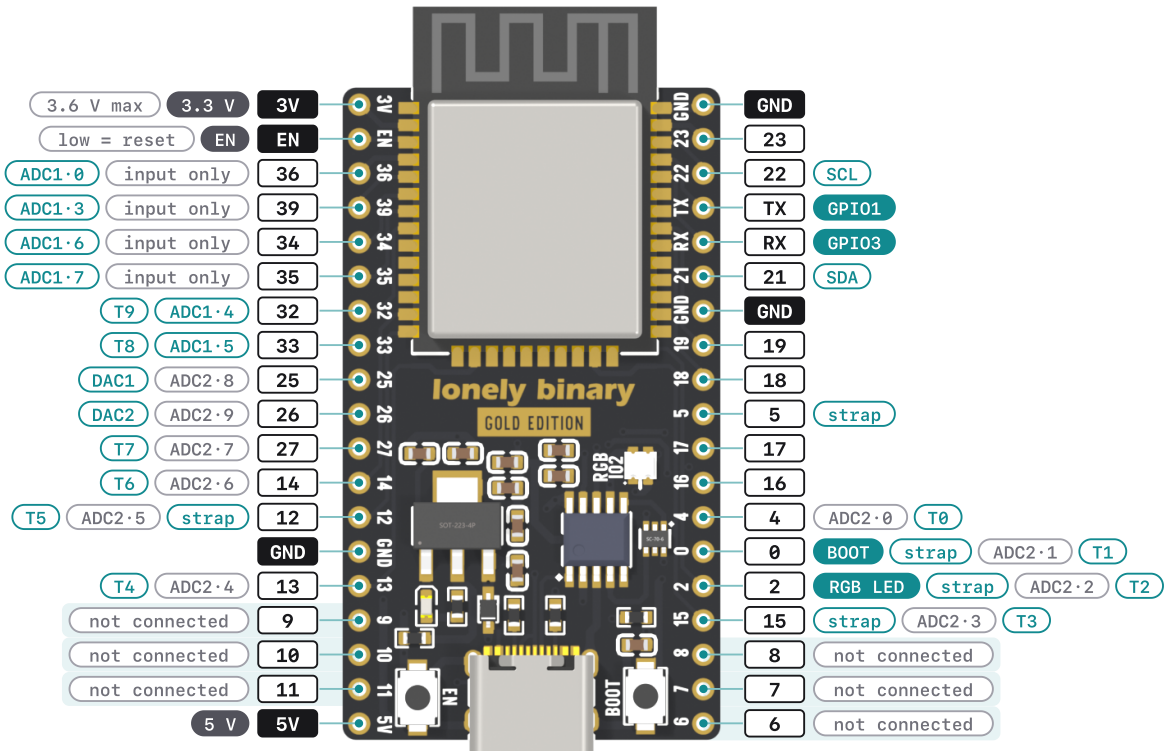


3.3 V logic

No GPIO takes 5 V, and the 3V pin takes 3.6 V at most. A lithium cell reads 4.2 V charged: feed it to 5V, never to 3V.

38 holes, as printed on the top

Antenna up, USB-C socket down. The back prints the same names, mirrored.



BOOT Already used by this board

not connected Drilled and labelled, wired to nothing

input only Reads only: no output, no pull-up

strap Read at reset. Keep quiet at power-up

ADC1·4 Analog input, works with Wi-Fi on

ADC2·0 Analog input, stops once Wi-Fi starts

Holes 6, 7, 8, 9, 10, 11

Drilled and labelled, and in no net. The chip's GPIO 6 to 11 still run its flash, so a sketch that uses those numbers still crashes, Serial1's default pins among them.

Read at reset

5

GPIO 0, 2, 5, 12, 15

GPIO12 high: no boot

Flashes over USB, will not start on its own supply? Almost always one of these.

Input only

4

GPIO 34, 35, 36, 39

no INPUT_PULLUP

A button here needs a real resistor. They are the best analog inputs: all ADC1.

Analog out

2

DAC on
GPIO 25, 26

dacWrite(25, 128)

8 bits, a real voltage rather than PWM. Both are ADC2, so no reads with Wi-Fi on.

First sketch: the RGB LED

Install FastLED by Daniel Garcia from the Library Manager first.

```
#include <FastLED.h>    // FastLED by Daniel Garcia
#define LED_PIN      2      // printed on the board as RGB IO2
#define NUM_LEDS     1
#define BRIGHTNESS   40    // 0-255; 255 is hard to look at

CRGB leds[NUM_LEDS];

void setup() {
  FastLED.addLeds<WS2812, LED_PIN, GRB>(leds, NUM_LEDS); // GRB
  FastLED.setBrightness(BRIGHTNESS);
  FastLED.clear();
  FastLED.show();
}

void flash(CRGB colour) {
  leds[0] = colour;
  FastLED.show();           // nothing changes until show()
  delay(1000);
  leds[0] = CRGB::Black;
  FastLED.show();
  delay(400);
}

void loop() {
  flash(CRGB::Red);
  flash(CRGB::Green);
  flash(CRGB::Blue);
}
```

red · green · blue, one second each
Red showing green means RGB where the sketch says GRB. The LED idles low, which is what the GPIO 2 strap needs.

When it does not work

It flashes, then will not start on its own

A strapping pin is held the wrong way at power-up, usually GPIO12 pulled high. Boot it bare, then add one wire at a time.

No port appears when it is plugged in

Try a cable that has moved a file: charge-only ones are common. Windows may also need the WCH CH340 driver.

A button on GPIO 34 reads at random

34, 35, 36 and 39 have no pull-up to switch on. Add a 10 kΩ resistor to 3V.

Three ways to power it

INTO	WHAT HAPPENS	WATCH FOR
USB-C	5 V from the cable, through a 1N5819WS Schottky, to the AMS1117-3.3	The 5V pin reads a little under the cable's 5 V. Normal
5V pin	Straight into the AMS1117-3.3; the diode stops it reaching the computer	Not a Vin pin: above about 5.5 V the extra is only heat
3V pin	Past the regulator, straight onto the module	3.6 V at most, and nothing on the board protects it

The regulator also feeds the module, the bridge and both LEDs. Budget about 500 mA for your own parts, less while the radio transmits.

Specifications

Module	ESP32-WROOM-32, 4, 8 or 16 MB flash, no PSRAM	CPU	dual-core Xtensa LX6, up to 240 MHz
Radio	2.4 GHz Wi-Fi, Bluetooth Classic and LE	USB	1 × USB-C, CH340K USB-to-serial bridge
RGB LED	WS2812B-2020 on GPIO 2	Regulator	AMS1117-3.3, 5 V in
Header	2 × 19 holes, 2.54 mm pitch, rows 25.40 mm apart	Board	48.20 × 27.90 mm, 1.6 mm thick

Arduino IDE settings

TOOLS MENU	SET TO
Board	ESP32 Dev Module
Port	The CH340K's port: wchusbserial on a Mac, a COM port on Windows
Upload Speed	921600, or 115200 if it stops part-way
Flash Size	4MB (32Mb), or the size on your module
Partition Scheme	Default 4MB with spiffs on a 4 MB module

Serial Monitor at 115200. The boot log arrives first at its own speed, so a line of symbols before your text is normal.