

Six SPI TFT display boards

0.96 to 3.5 inch. One circuit on six outlines: the same eight holes, the same 15-way socket, one library.



3.3 V only

CS, DC, RST, SCL and MOSI go straight to the controller, with no resistor and no level shifter, and no regulator sits behind the 3V3 hole. **No 5 V on any of them.** A 5 V board needs a level converter on those five lines and a 3.3 V rail for the 3V3 hole.

One row of eight holes, on all six

Drawn on the 2.4 inch. The other five carry the same eight holes, the same names and the same 15-way socket: only the outline and the panel socket change. Hole 1 is GND, at the left end seen from the front.

	1	2	3	4	5	6	7	8
Front	GND	3V3	CS	MOSI	SCL	DC	RST	BL
15-way pin	1	2	9	8	7	5	4	3
ESP32-S3	GND	3V3	10	11	12	2	42	41
ESP32	GND	3V3	15	23	18	2	4	32

Jumper BL to 3V3

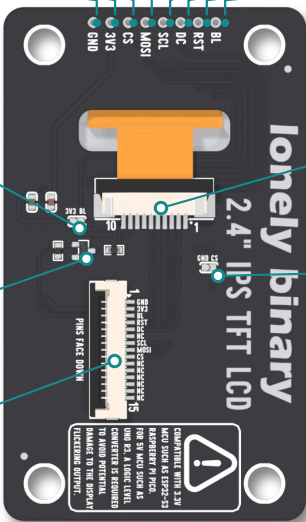
Open as shipped. Bridged: lamp always on. Leave BL unwired.

Backlight switch Q2

BSS138, in the lamp's low side. 10 kΩ holds it off until BL is HIGH.

15-way socket U31

1.0 mm, pins face down. The same eight nets as the row.



Panel socket U32, 10 way
1.0 mm pitch, one per board.
The glass plugs here, not U31.

Jumper CS to GND
Open as shipped. Bridged: CS is held low. Leave CS unwired.

Dark ≠ dead

A black screen with a good picture on it is a backlight nobody drove, not a dead panel. A 10 kΩ resistor holds the lamp off until BL is driven (LOW on the 0.96 inch, HIGH on the other five): tilt the screen under a bright lamp and the picture is there.

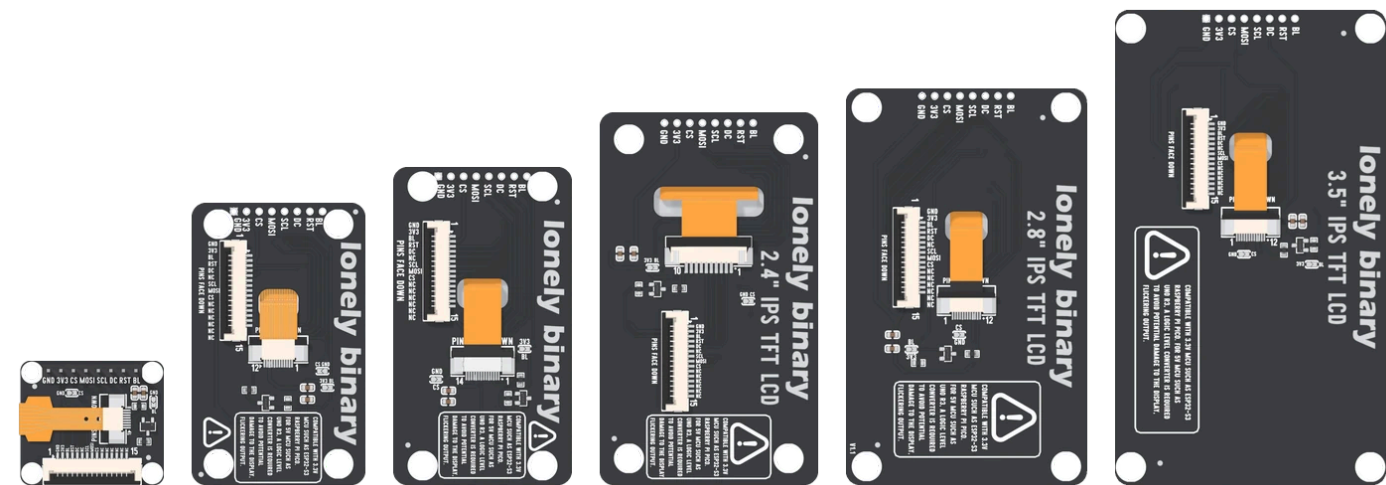
Six boards, side by side

BOARD	CONSTANT	DRIVER	PIXELS	CLOCK	REDRAW, AT LEAST	
0.96 inch	LB_TFT_096	ST7735	80 × 160	20 MHz	10.2 ms	
1.8 inch	LB_TFT_18	ST7735	128 × 160	20 MHz	16.4 ms	
2.0 inch	LB_TFT_20	ST7789	240 × 320	40 MHz	30.7 ms	
2.4 inch	LB_TFT_24	ST7789	240 × 320	40 MHz	30.7 ms	
2.8 inch	LB_TFT_28	ST7789	240 × 320	40 MHz	30.7 ms	
3.5 inch	LB_TFT_35	ST7796	320 × 480	40 MHz	61.4 ms	

Redraw is arithmetic, not a measurement: pixels × 2 bytes × 8 bits ÷ the SPI clock in the library's panel table. No board was timed on this machine, and neither an ESP32-S3 nor a classic ESP32 was tested. Real redraws are slower. Bars share one scale: full length is 125 ms.

Six outlines, one circuit

Every board carries the same eight holes, the same 15-way socket, the same backlight arrangement and two solder jumpers. What the glass decides is the outline and the panel socket.



Which board

BOARD	OUTLINE, MM	PANEL SOCKET	LAMP SWITCH, LIT WHEN	GLASS, MM	PX/IN
0.96 inch	29.2 × 25.0	U32, 8 way, 0.5 mm	AO3401, BL LOW	10.8 × 21.7	188
1.8 inch	35.0 × 56.7	U7, 12 way, 0.5 mm	BSS138, BL HIGH	28.032 × 35.04	116
2.0 inch	36.0 × 64.0	FPC1, 14 way, 0.5 mm	BSS138, BL HIGH	30.6 × 40.8	199
2.4 inch	44.0 × 75.0	U32, 10 way, 1.0 mm	BSS138, BL HIGH	36.72 × 48.96	166
2.8 inch	48.6 × 79.8	FPC1, 12 way, 0.5 mm	BSS138, BL HIGH	43.2 × 57.6	141
3.5 inch	54.6 × 95.6	FPC2, 12 way, 0.5 mm	BSS138, BL HIGH	48.96 × 73.44	166

The size on the box decides it: the pixels follow the size, and the code follows the constant. Glass is the active area from the panel makers' sheets; px/in is worked out from it and the pixel count, not a datasheet figure. No board has touch or a card socket. Tinted rows are the two boards that differ from the other four in one way, below.

What only one board has

0.96 inch: backlight and offset

Its backlight switch is a P-channel AO3401 in the lamp's supply, so BL LOW lights the lamp and BL HIGH or open holds it off. The other five are the other way up. Its bridge pair is BL to GND, not 3V3. The library's panel table also gives this glass a column offset of 24, so a hand-written setup draws the picture shifted. The library carries both, and nothing in your sketch changes.

2.4 inch: a 1.0 mm panel socket

U32 is a 10-way FPC socket at 1.0 mm pitch. The other five take 0.5 mm tails, 8, 12, 14, 12, 12 ways, so a tail cut for 0.5 mm cannot seat in it.

Your first picture, any board

```
/* First light: any of the six. Arduino IDE settings
   Board: ESP32S3 Dev Module, USB CDC On Boot: Enabled
   (or ESP32 Dev Module on a classic ESP32), Serial 115200 */
#include <LonelyBinaryDisplay.h>

// One constant per board, the one line that changes:
// LB_TFT_096 LB_TFT_18 LB_TFT_20 LB_TFT_24 LB_TFT_28 LB_TFT_35
LB_Display display(LB_TFT_24);

void setup() {
  Serial.begin(115200);
  delay(500);
  if (!display.begin()) {
    Serial.println("begin() failed: SPI bus did not start");
    return;
  }
  Serial.printf("%d x %d\n", display.width(), display.height());
  display.fillScreen(LB_BLACK);
  display.setTextColor(LB_WHITE);
  display.setTextSize(2);
  display.drawString("Hello", 8, 8);
  display.flush(); // always safe to call
}

void loop() {}
```

240 x 320

What Serial Monitor at 115200 should print: the size width() and height() report, which is your board's own. The screen shows Hello in white on black.

To change the glass, change one token

BOARD	CONSTANT	PIXELS
0.96 inch	LB_TFT_096	80 × 160
1.8 inch	LB_TFT_18	128 × 160
2.0 inch	LB_TFT_20	240 × 320
2.4 inch	LB_TFT_24	240 × 320
2.8 inch	LB_TFT_28	240 × 320
3.5 inch	LB_TFT_35	320 × 480

When it does not work

The screen is dark but a picture shows at an angle

The panel is drawn on and the lamp is off. BL is unwired, never driven, or set to zero; the 10 kΩ holds the lamp off until it is. Check BL driven (LOW on the 0.96 inch, HIGH on the other five). The other seven holes are fine.

begin() returns false

The SPI bus itself would not start, which is almost always a pin number the chip does not have (a C3 on the defaults for the other two). A wiring fault on DC, RST or CS cannot make it return false: the panel is written to and never read, so begin() returns true with a blank screen.

It flickers, or the picture tears halfway down

The signal arrives but not cleanly: wires too long, the clock too fast, or a 5 V board on a 3.3 V input. Shorten the wires, call setSpiHz(2000000) before begin(), and check the logic level of the host.

Specifications

Supply	3.3 V only. No regulator, no level shifter
15-way	15 ways, 1.0 mm FPC (U31)
Clock	20 MHz on the ST7735 boards, 40 MHz on the rest
Outline	29.2 × 25.0 mm to 54.6 × 95.6 mm, 1.6 mm board

Wiring

HOLE	NAME	BACK	15-WAY	ESP32-S3	ESP32
1	GND	8	1	GND	GND
2	3V3	7	2	3V3	3V3
3	CS	6	9	10	15
4	MOSI	5	8	11	23
5	SCL	4	7	12	18
6	DC	3	5	2	2
7	RST	2	4	42	4
8	BL	1	3	41	32

The same on all six boards: the pins belong to the host and the glass to the constant. Both faces print the same names; **Back** counts the holes from BL, as the back reads. These are the library's defaults: setWiring() before begin() changes them.

The two solder jumpers

BOARD	TIES CS TO GND	TIES BL TO A RAIL
0.96"	CS – GND	BL – GND
1.8"	CS – GND	BL – 3V3
2.0"	CS – GND	BL – 3V3
2.4"	CS – GND	BL – 3V3
2.8"	CS – GND	BL – 3V3
3.5"	CS – GND	BL – 3V3

Both pairs ship open. Bridged, the CS pair holds CS low (always selected, no other device on the bus) and the BL pair lights the lamp for good; the 0.96 inch ties BL to GND, the others to 3V3. Do not drive a bridged pad: the pin would fight a rail.

It is plain white and has never shown anything

Many panels power up white: the controller is alive and nothing has reached it. Check DC, then RST, then the 15-way cable if you use it, contacts down. Power, ground and the lamp are proven.

Red comes out blue, or it is a photo negative

Colour order and inversion are one call each: setColorOrder() and setInverted(), before or after begin(). Both at once turns red into yellow. The right values belong in the library's table: report them.

Nothing works on an ESP32-C3

The library's default pins are for the ESP32-S3 and the classic ESP32; GPIO 23 and 32 do not exist on a C3. Copy LB_WIRING, change the pins to ones the C3 has, and call setWiring() before begin().