LESSON · GUITION 4.3" AND 7" ESP32-P4 DISPLAYS

The Guition ESP32-P4 displays

Revision 1.2 · 21 November 2026

Guition's ESP32-P4 boards are touch screens with a fast processor, plenty of memory, Wi-Fi and a speaker, in a case ready to hang on a wall or stand on a bedside table. This lesson takes you round the 4.3" and the 7", shows how SimuCraft's four examples for them are built, and passes on what we learned putting them on real boards: the pitfalls, and how to get past each one.

What you need

- **SimuCraft for Windows**, 0.6.5 or later. Steps 1 to 5 also work in the web version; uploading needs the desktop app.
- For steps 6 to 8: a Guition **JC4880P443C** (4.3") or **JC1060P470C** (7"), and a USB-C cable that carries data.
- For the wall panel on your own house: **Home Assistant**. For the others, only Wi-Fi.
- About an hour, plus one long first build (see Why the first build is slow).

1. The boards

Both boards have the same heart, an ESP32-P4 with an ESP32-C6 beside it for the radio, and differ in the screen:

	4.3" JC4880P443C	7" JC1060P470C
Screen	480 × 800 IPS, ST7701S chip; our examples turn it to landscape, 800 × 480	1024 × 600 IPS, JD9165 chip, landscape as it lies
Touch	GT911 on I2C (SDA 7, SCL 8), reset GPIO 3	GT911 on I2C (SDA 7, SCL 8), reset GPIO 22
Processor	ESP32-P4: two RISC-V cores (360 MHz on the chips sold in 2026), 32 MB PSRAM, 16 MB flash	
Wi-Fi and Bluetooth	A second chip, an ESP32-C6, on its own link: the P4 has no radio. Sketches use `WiFi.h` as usual.	
Sound	ES8311 codec and a speaker amplifier (speaker connector on the back)	
Backlight	GPIO 23, dimmable with PWM	

The screens are joined to the P4 by **MIPI-DSI**, the fast serial link phones use. The P4 keeps a copy of the whole picture in PSRAM and streams it to the screen sixty times a second; your sketch only

changes the copy.

In SimuCraft, both are one part: **Guition ESP32-P4 display**, with **Board** in Properties set to 4.3" or 7". Turn it over with the round arrows to see the back: the 2 × 13 header, the USB-C sockets and the buttons.

2. Ports and buttons

- **FULL SPEED USB-C**: power, uploads and the Serial Monitor. Always plug in here to upload.
- **HIGH SPEED USB-C**: the P4's fast USB, for USB devices. Not for uploading.
- Three buttons on the top edge. Seen from the front, the one on its own is the **battery key** (it switches a battery pack on and off); of the pair, the outer one is **BOOT** and the inner one is **RESET**.
- **BOOT** is GPIO 35. Held while the board powers up, it stops the sketch and waits for an upload. After that it is free: Hello reports each press on the screen and the Serial Monitor.
- **RESET** restarts the P4, and with it your sketch.

3. Hello: how a sketch drives the screen

Open **Examples** → **Guition 4.3" P4 display** (or 7") → **Hello**. Tap the button to count; drag the slider to dim the screen. The sketch is short, and every graphics sketch for these boards has the same four parts:

1. **The panel**: an `Arduino_ESP32DSIPanel` with the screen's timings, and an `Arduino_DSI_Display` with its size, its reset pin and its **start-up table**: the list of commands the screen's chip needs before it shows anything (from GFX Library for Arduino).
2. **LVGL**, the graphics library that draws buttons, sliders and text. It draws into a buffer in PSRAM.
3. `flush_cb`: LVGL calls it with each finished area, and it copies the pixels to the panel.
4. **The touch read**: the GT911's position, turned the same way as the picture, handed to LVGL as its pointer.

Everything else, the button and the slider, is ordinary LVGL. The 7" Hello differs only in the panel part: its JD9165 screen, its timings and its table.

4. The designed examples

The other three examples are made in the **screen designer**: click **Design screen** on the board to see the screens, move things, change fonts and colours. The designer writes the sketch, in sections:

- **The board's set-up** (the same four parts as Hello), written by the designer.
- **The screen's widgets**, named `ui_…`, and `ui_create()`, which builds them.

- **Your own code**, between `// USER CODE BEGIN` and `END` lines: *globals* (your variables and functions), *setup*, *loop*, and one for each button's or slider's action (`ui_next_clicked`, `ui_arc_changed`, ...). The designer keeps these when it writes the sketch again.
- **Pictures and fonts** in `ui_assets.h`, a file saved beside the sketch.

The three examples:

- **Bedside clock radio**: huge thin figures, the date, the weather now and five days ahead. At night the screen drops to a glow, and a first touch only wakes it; the brightness is set in the designer (the screen's **Brightness**) and on the board's Setup screen. An alarm (kept with `Preferences` when the power is off) brightens the screen like a sunrise and wakes you to internet radio through the board's speaker, using the ESP32-audioI2S library. Its main screens (the full set of ten, and its code step by step, are in The clock radio):
  - **Clock**, with the alarm, the Radio button and the weather.
  - **Alarm**: on or off, the hour and minute.
  - **Radio** (the Radio button): the station and the song, play, previous and next, a favourite star, a level meter, the volume, a sleep timer, your stations (favourites first) and **Find stations**, which searches radio-browser.info's free directory with the on-screen keyboard and keeps what you pick.
  - **Weather** (tap the weather, or Forecast ›): the weather now, humidity, wind with a compass arrow, pressure, UV, rain, sunrise and sunset, and the week.
  - **Setup** (the gear): brightness by day and by night, when night starts and ends, dimming when untouched, Wi-Fi, your town and the alarm's loudest volume.
- **Weather station**: the weather now, six tiles (humidity, wind, pressure, UV, rain, sun), the week ahead, and Next 24 h. Change the town in Design screen → Wi-Fi and weather. Step by step in The weather station.
- **Smart-home wall panel**: four lights, a thermostat arc, the blinds, the garage door, solar and battery power, and two scenes, all through Home Assistant. Step by step in The Home Assistant wall panel.

How the designer's part and your own fit together, in detail, is the subject of Writing for the display boards. Your code in these sketches uses the designer's names freely: `ui_set` for the thermostat's label, `ui_lat` for the town's latitude. When the sketch goes to the board, SimuCraft puts those declarations above your code, where C++ needs them (see What SimuCraft puts in order for you).

5. Wi-Fi, weather and Home Assistant

- **Wi-Fi**: the upload window asks for your network's name and password and puts them in the sketch it sends. On the board, tapping the Wi-Fi icon opens a Wi-Fi screen: the networks it can see, then a keyboard across the screen for the password (**ABC** for capitals, **1#** for figures and symbols, Show to check it); the board keeps the network for next time. In SimuCraft the boards join the simulated lab network.

- **The time** comes from the internet's time servers once Wi-Fi is up; until then the clock shows dashes.
- **The weather** comes from Open-Meteo, free and with no key. The weather station also reads the nearest airport's own report (its METAR) for the temperature and wind, which near an airport is closer than a forecast.
- **Home Assistant**: the wall panel talks to its REST API. In SimuCraft a pretend house answers at `homeassistant.local:8123`, to any token: until one is typed in, the panel asks nothing and says so. For your own, put your Home Assistant's address and a **long-lived access token** (your profile → Security) at the top of the sketch, and your own entity names (`light.lounge`, `climate.house`, ...). The panel reads the house every 3 seconds, so changes made in the Home Assistant app show on the wall too.

6. Uploading

1. Plug the **FULL SPEED** socket into your PC with a data cable. Windows shows a new COM port (*USB Serial Device*; in Device Manager, under Ports).
2. In SimuCraft click **Real board** (or Upload on the board). Choose the port and the board (4.3" or 7"; SimuCraft offers the right one for the part), and type your Wi-Fi name and password.
3. Click **Upload to the board**. The first build takes 5 to 10 minutes; later ones under a minute.
4. SimuCraft checks the P4's **chip revision** as it uploads. Boards sold in 2026 have revision 1.3 ("before v3.00"); if yours is newer, it builds again for it, by itself, and remembers.
5. The board restarts with your sketch. Messages from `Serial.println` appear in the window's Serial Monitor.

7. Pitfalls

The 7" has two panels

Guition changed the 7"'s screen in 2026. The new one (Guition and the community call it **V2**) needs its own start-up table, a screen reset on GPIO 0, and 52 MHz timings; boards may carry a "V2" label, but ours had none. With the first panel's settings a V2 screen shows a grey or white band with vertical stripes, or stays blank. SimuCraft's 7" examples use the V2 settings. One detail cost us a day: the GFX library's 7" table ends with a "16-bit colour" command (`0x3A`) that Guition's own code leaves out. With it, the picture is squashed into the middle half of the screen.

A stuck picture

An LCD driven with the wrong voltages for a while keeps a ghost of what it showed, and stripes where it was stressed. It isn't broken: **unplug the board for a few hours** (no USB, no battery), then run a sketch with the right settings. Ours cleared within the hour, running a correct sketch.

The COM port keeps appearing and disappearing

The sketch is crashing and restarting, and the USB port goes with it each time. Unplug the cable, **hold BOOT**, plug the cable back in, and let go after two seconds: the board waits in upload mode, the port stays, and you can upload a working sketch.

"Port busy" or no port

Something else has the port open: SimuCraft's Serial Monitor, the Arduino IDE, or a browser tab with a web installer. Close it, unplug, plug in again. If no port appears at all, try another cable: many USB-C cables carry power only.

A pale picture

With the backlight at full, dark backgrounds look grey-blue. Bring the slider down a little: blacks get blacker.

A label shows "f" or "f mm"

The LVGL built for these boards has no floating-point printing, so a `%f` or `%.1f` in `lv_label_set_text_fmt` prints just the letter f. Make the text with `snprintf` first, then set it. SimuCraft's simulator shows the f too, and Verify warns about it (see Decimals: never %f).

"Sketch too big"

The clock radio, with its fonts, pictures and the audio library's decoders, needs more than the 3 MB the usual ESP32 layouts give a program. SimuCraft gives the P4 boards a 7 MB program space of their 16 MB flash.

8. Tips

- **Print as you go.** A `Serial.println` in `setup()` and every few seconds in `loop()` tells you the sketch is running even when the screen isn't.
- **Power-cycle after changing the screen's settings.** The panel keeps its old settings through a RESET or an upload; only a full power-off clears them.
- **Use BOOT as a button.** GPIO 35 with `pinMode(35, INPUT_PULLUP)`: LOW while pressed.
- **Draw in PSRAM.** These boards have 32 MB: give LVGL a full-screen buffer there, as our examples do, for smooth redraws.
- **Dim at night.** Select the screen in the designer and tick **Brightness → The designer handles it**: the night hours, a low glow, a floor, and a first touch that only wakes it, with no code. Your code can take over with `ui_bl_force`, as the clock does for its alarm.

Things to try

- Make BOOT switch the bedside clock between day and night mode.
- Add a fifth light to the wall panel: one more tile in Design screen, one more entity name in the code.
- Put the weather station's town on a button, so a tap switches between two places.
- Run the same designed screen on both sizes: copy it, change Board in Properties, and see what Design screen makes of the bigger screen.

Revision history

Revision	Date	What changed
1.0	16 November 2026	First edition.
1.1	18 November 2026	The clock radio's five screens (radio with station search, weather, setup); the Wi-Fi screen and its keyboard.
1.2	21 November 2026	The clock's brightness and the alarm's off look are now set in the designer; the night look (a glow, no longer red figures); the "f" pitfall; links to the new lessons on writing for the display boards and the three flagship examples; the wall panel needs a token, any token, in SimuCraft too.