

ESP32-S3-Touch-LCD-4.3B pin map

This document summarises the I/O assignments for **Waveshare's ESP32-S3-Touch-LCD-4.3B** development board. The board uses an **ESP32-S3-WROOM-1-N16R8** module (16-MB flash & 8-MB PSRAM) and a 4.3-inch 800×480 RGB display. Most of the microcontroller's pins are used by the display, touch interface, USB-OTG and communications interfaces, so only a few pins remain free. Digital inputs/outputs are isolated through opto-couplers and controlled via the **CH422G** I²C GPIO expander. The pin-maps below are verified from the Waveshare schematic and wiki ¹.

On-board interfaces and major components

- **Display:** 24-bit RGB interface driven by **ST7262** LCD driver. Signals such as pixel clock (PCLK), horizontal/vertical sync (HSYNC/VSYNC) and 8-bit red/green/blue data buses use many GPIO pins ². A boost converter (**MP3302**) provides the LCD back-light; the back-light enable (DISP) and LCD reset lines are controlled by the CH422G expander.
- **Touch panel:** Capacitive touch controller **GT911** connected via I²C. The microcontroller uses **GPIO8** and **GPIO9** for the I²C bus; **GPIO4** is the touch interrupt line; the **CH422G** controls the touch reset signal ³.
- **USB-OTG:** The on-chip USB interface uses **GPIO19** (D-) and **GPIO20** (D+) ⁴. A USB-switch (**FSUSB42**) controlled by **EXIO5** chooses between the USB port and the CAN transceiver.
- **Micro-SD (TF) card:** Connected via SPI; **GPIO11** (MOSI), **GPIO12** (SCK) and **GPIO13** (MISO) are used for the SPI bus, and the SD-card chip-select is controlled by **EXIO4** ⁵.
- **CAN bus:** NXP **TJA1051** transceiver. The microcontroller's **GPIO15** (CANTX) and **GPIO16** (CANRX) provide TX and RX for the on-chip TWAI controller ⁶. The transceiver is connected through the FSUSB42 switch so that CAN and USB share the D+/D- lines.
- **RS-485:** MAX3485-compatible transceiver (**SP3485EN**). **GPIO43** and **GPIO44** provide RS-485 RXD and TXD ⁷.
- **Isolated digital I/O:** Two opto-isolated digital outputs (**DO0/DO1**) and two opto-isolated digital inputs (**DI0/DI1**) support 5–36 V levels and up to ~450 mA sink current ⁸. These lines are controlled by the CH422G expander (EXIO0 = DI0, EXIO5 = DI1, OD0 = DO0, OD1 = DO1) ⁹.
- **RTC:** Real-time-clock chip **PCF85063** uses the same I²C bus (GPIO8/GPIO9) plus an interrupt pin connected to **GPIO6**.
- **Other devices:** USB-to-UART (**CH343P**), CAN/RS-485 terminal resistor switch, battery charger (**CS8501**), a multi-channel IO expander (**CH422G**), and voltage regulators (SGM2212 etc.) ¹⁰.

P1 external terminal pinout

The board exposes power and bus lines on a 16-pin **PORT1 (P1)** terminal block. Pin 1 is nearest the VIN label on the silk screen. VOUT and I²C pull-up voltage may be configured for 5 V or 3.3 V via onboard jumpers ¹¹.

P1 pin	Signal	Description
1	VIN	External DC supply (7 – 36 V).
2	GND	Ground reference.
3	I2C_VCC	I ² C pull-up voltage (selectable 3.3 V/5 V) ¹¹ .
4	GND	Ground.
5	D_SDA	External I ² C data; level shifted to/from I2C_VCC. Connected to ESP32-S3 GPIO8 through a level shifter.
6	D_SCL	External I ² C clock; level shifted. Connected to ESP32-S3 GPIO9.
7	CANL	CAN bus low line.
8	CANH	CAN bus high line.
9	RS485_TX–	RS-485 differential transmit negative line.
10	RS485_TX+	RS-485 differential transmit positive line.
11	DOUT0	Opto-isolated digital output 0 (open-drain, 5–36 V, sink up to 450 mA) ⁸ . Controlled by CH422G OD0.
12	DOUT1	Opto-isolated digital output 1 (controlled by CH422G OD1).
13	DI_COM	Common terminal for opto-isolated digital inputs. Leave floating for dry contact input; tie to external supply positive/negative for active sensing ¹² .
14	GND	Ground.
15	DIN0	Opto-isolated digital input 0 (connected to CH422G EXIO0).
16	DIN1	Opto-isolated digital input 1 (connected to CH422G EXIO5).

ESP32-S3 WROOM-1 pin mapping

The following table lists each ESP32-S3 GPIO on this board, the primary board-level function and the underlying MCU capabilities. Because the LCD consumes most pins, very few are available for other uses. Pins shown as **unused** are not connected to external circuits but may be tied internally or used as strapping pins; repurposing them usually requires modification of the board.

GPIO	Board use	MCU capabilities ¹	Notes & repurposing
GPIO0	G3 (LCD green data bit 3) ²	ADC1_CH0; digital I/O; PWM	Used for LCD bus; also a strapping pin determining boot mode, so repurposing is not recommended.
GPIO1	R3 (LCD red data bit 3) ²	ADC1_CH1; digital I/O; PWM	Part of RGB bus; not available for user I/O.

GPIO	Board use	MCU capabilities ¹	Notes & repurposing
GPIO2	R4 (LCD red data bit 4) ²	ADC1_CH2; digital I/O; PWM	LCD bus.
GPIO3	VSYNC (vertical sync) ²	ADC1_CH3; digital I/O; PWM; strapping pin	Used for LCD synchronisation and bootstrapping; avoid reuse.
GPIO4	TP_IRQ (touch interrupt) ³	ADC1_CH4; digital I/O; PWM	Interrupt from GT911. Could be repurposed only if the touch controller is unused.
GPIO5	DE (LCD data-enable) ²	ADC1_CH4; digital I/O; PWM	Required for LCD. On the ESP32-S3, this pin maps to ADC1 channel 4 (channel 5 is actually on GPIO6).
GPIO6	RTC_INT (PCF85063 interrupt)	ADC1_CH5; digital I/O; PWM	Provides the RTC alarm interrupt. On the ESP32-S3 this pin corresponds to ADC1 channel 5. If the RTC interrupt isn't needed, this pin can be used as a regular GPIO.
GPIO7	PCLK (LCD pixel clock) ²	ADC1_CH6; digital I/O; PWM	Drives the LCD.
GPIO8	I²C SDA – bus shared by touch controller, CH422G expander and RTC ¹³	ADC1_CH7; digital I/O; PWM; I ² C SDA	Provides master I ² C data for board peripherals and external I ² C port.
GPIO9	I²C SCL – bus clock ¹³	ADC1_CH8; digital I/O; PWM; I ² C SCL	Same as above.
GPIO10	B7 (LCD blue data bit 7) ²	ADC1_CH9; digital I/O; PWM	Part of RGB bus.
GPIO11	MOSI (SPI MOSI for TF card) ⁵	ADC2_CH0; digital I/O; PWM; SPI2 MOSI	Used by the micro-SD card; can be used for SPI if the card is unused.
GPIO12	SCK (SPI clock for TF card) ⁵	ADC2_CH1; digital I/O; PWM; SPI2 CLK	Used by the micro-SD card.
GPIO13	MISO (SPI MISO for TF card) ⁵	ADC2_CH2; digital I/O; PWM; SPI2 MISO	Used by the micro-SD card.
GPIO14	B3 (LCD blue data bit 3) ²	ADC2_CH3; digital I/O; PWM	Part of RGB bus.

GPIO	Board use	MCU capabilities ¹	Notes & repurposing
GPIO15	CANTX – CAN bus transmit ⁶	ADC2_CH4; digital I/O; PWM; UART TX; CAN_TX	Used by TWAI controller via TJA1051. If CAN bus is not used, this pin can be repurposed as a UART-TX or general output, but note that the USB/CAN multiplexer and TJA1051 remain connected.
GPIO16	CANRX – CAN bus receive ⁶	ADC2_CH5; digital I/O; PWM; UART RX; CAN_RX	As above. Can be used for UART-RX if CAN is disabled.
GPIO17	B6 (LCD blue data bit 6) ²	ADC2_CH6; digital I/O; PWM	RGB bus.
GPIO18	B5 (LCD blue data bit 5) ²	ADC2_CH7; digital I/O; PWM	RGB bus.
GPIO19	USB_DN (USB D- line) ⁴	ADC2_CH8; digital I/O (no ADC when USB enabled); USB_OTG D-	Dedicated to on-chip USB; not available as GPIO when USB is used.
GPIO20	USB_DP (USB D+ line) ⁴	ADC2_CH9; digital I/O (no ADC when USB enabled); USB_OTG D+	Same as above.
GPIO21	G7 (LCD green data bit 7) ²	Digital I/O; PWM	RGB bus.
GPIO38	B4 (LCD blue data bit 4) ²	Digital I/O; PWM	RGB bus.
GPIO39	G2 (LCD green data bit 2) ²	Digital I/O; PWM	RGB bus.
GPIO40	R7 (LCD red data bit 7) ²	Digital I/O; PWM	RGB bus.
GPIO41	R6 (LCD red data bit 6) ²	Digital I/O; PWM	RGB bus.
GPIO42	R5 (LCD red data bit 5) ²	Digital I/O; PWM	RGB bus.
GPIO43	RS485_RXD (RS-485 receive) ⁷	Digital I/O; UART RX	Used by the SP3485 transceiver. Can be repurposed as UART-RX if RS-485 is unused.

GPIO	Board use	MCU capabilities ¹	Notes & repurposing
GPIO44	RS485_TXD (RS-485 transmit) ⁷	Digital I/O; UART TX	Used by the SP3485 transceiver; can be used as UART-TX when RS-485 is disabled.
GPIO45	G4 (LCD green data bit 4) ²	Digital I/O; PWM; strapping pin	LCD bus and boot strapping; do not repurpose.
GPIO46	HSYNC (LCD horizontal sync) ²	Digital I/O; PWM; strapping pin	LCD bus and boot strapping; do not repurpose.
GPIO47	G6 (LCD green data bit 6) ²	Digital I/O; PWM	RGB bus.
GPIO48	G5 (LCD green data bit 5) ²	Digital I/O; PWM	RGB bus.
EXIO0	DIO – opto-isolated digital input 0 ⁹	I ² C-controlled via CH422G	Provides isolated input sensing (5–36 V). Not a direct ESP32 pin.
EXIO1	TP_RST – touch panel reset ³	I ² C-controlled via CH422G	Drives GT911 reset.
EXIO2	DISP – LCD/back-light enable ¹⁴	I ² C-controlled via CH422G	Enables the LCD and its boost converter.
EXIO3	LCD_RST – LCD reset line	I ² C-controlled via CH422G	Resets the ST7262 LCD driver.
EXIO4	SD_CS – micro-SD card chip select ⁵	I ² C-controlled via CH422G	Active low; required for TF card access.
EXIO5	DI1 – opto-isolated digital input 1 ⁹ and USB/CAN switch control	I ² C-controlled via CH422G	Reads external digital input; also controls the FSUSB42 switch selecting USB vs. CAN.
OD0	DO0 – opto-isolated digital output 0 ⁹	I ² C-controlled via CH422G	Open-drain output; sinks up to ~450 mA, 5–36 V.
OD1	DO1 – opto-isolated digital output 1 ⁹	I ² C-controlled via CH422G	Same as above.

¹MCU capabilities column lists standard ESP32-S3 features for each GPIO: ADC channels (ADC1/ADC2), PWM (LEDC), I²C, SPI or UART functions. When pins are used for dedicated board functions (LCD, USB, etc.), those capabilities are usually unavailable or require hardware modification.

Unused or re-purposable pins

The 4.3-inch RGB display occupies almost all of the ESP32-S3's GPIO lines, leaving very little I/O for user-defined functions. There are no free pins broken out on this board. However, the following possibilities exist if certain peripherals are not used:

- **RS-485 lines (GPIO43 & GPIO44):** If the RS-485 port is unused, these pins can be reconfigured as a UART or general digital I/O. They are not ADC capable. Removing the SP3485 or disconnecting its enable line may be necessary.
- **CAN lines (GPIO15 & GPIO16):** If the CAN bus is not used and USB is preferred, the FSUSB42 switch can be left in USB mode. In that case these pins may be used as UART TX/RX or PWM outputs. Be aware that they are still connected to the TJA1051 transceiver and may require tri-stating.
- **Micro-SD SPI pins (GPIO11-13 & EXIO4):** If no TF card is used, these pins can become a general-purpose SPI bus (HSPI). EXIO4 must be set high (chip select inactive) via the CH422G.
- **Touch interrupt (GPIO4):** If the touch controller is not used, GPIO4 can be repurposed as an ADC input or digital output.
- **RTC interrupt (GPIO6):** If the real-time-clock alarm feature is unnecessary, GPIO6 is available as an ADC input or general digital pin.
- **Unused ADC channels:** Although LCD pins are connected to the RGB bus, the underlying ADC channels on GPIO0–GPIO10 may still work for low-frequency measurements; however, the LCD driver toggles these lines continuously, so they are not recommended for analog use.
- **CH422G lines:** Unused expander pins (e.g., DI1, DO0, DO1) can be controlled from software for custom signalling up to 36 V. Additional CH422G pins not brought out are not accessible without hardware changes.

Notes on boot-strapping and restrictions

- **Strapping pins:** GPIO0, GPIO3, GPIO45 and GPIO46 are used as strapping pins that select the boot mode or the flash voltage during reset. They are also tied to the LCD interface on this board. Avoid driving them externally during power-up.
- **USB function pins:** GPIO19 and GPIO20 are dedicated to the ESP32-S3's USB-OTG peripheral ⁴. They cannot be used as general I/O when the USB peripheral is enabled.
- **RS-485/CAN termination:** A DIP switch on the board allows the 120 Ω termination resistor for CAN/RS-485 to be inserted or removed ⁷. This should be set according to the bus topology.
- **Digital I/O isolation:** The isolated digital inputs and outputs use opto-couplers and can handle higher voltages, but they are controlled via the CH422G expander. They are not directly connected to ESP32 pins ⁹.

Summary

The ESP32-S3-Touch-LCD-4.3B is optimised for display-centric applications and exposes only a handful of spare GPIOs. All essential interfaces—display, touch, USB, micro-SD, CAN, RS-485, I²C and isolated digital I/O—are pre-wired on the board. Careful planning is required if your application needs additional pins; consider disabling unused peripherals or using the CH422G expander lines to accommodate extra signals.

