ESP32-P4

Hardware Design Guidelines

Release master
Espressif Systems
Jun 06, 2025

Table of contents

Table of contents i

1 Latest Version of This Document 3
- 1.1 About This Document . . . 3
  - 1.1.1 Introduction . . . 3
  - 1.1.2 Latest Version of This Document . . . 3
- 1.2 Product Overview . . . 3
- 1.3 Schematic Checklist . . . 4
  - 1.3.1 Overview . . . 4
  - 1.3.2 Power Supply . . . 4
  - 1.3.3 Chip Power-up and Reset Timing . . . 8
  - 1.3.4 Flash and PSRAM . . . 9
  - 1.3.5 Clock Source . . . 10
  - 1.3.6 UART . . . 11
  - 1.3.7 SPI . . . 11
  - 1.3.8 Strapping Pins . . . 11
  - 1.3.9 GPIO . . . 12
  - 1.3.10 ADC . . . 14
  - 1.3.11 SDIO . . . 14
  - 1.3.12 USB . . . 15
  - 1.3.13 Touch Sensor . . . 16
  - 1.3.14 Ethernet MAC . . . 16
  - 1.3.15 MIPI . . . 16
- 1.4 PCB Layout Design . . . 16
  - 1.4.1 General Principles of PCB Layout for the Chip . . . 17
  - 1.4.2 Crystal . . . 17
  - 1.4.3 USB . . . 18
  - 1.4.4 SDIO . . . 18
  - 1.4.5 Touch Sensor . . . 19
  - 1.4.6 MIPI . . . 21
- 1.5 Hardware Development . . . 22
  - 1.5.1 ESP32-P4 Development Boards . . . 22
  - 1.5.2 Download Guidelines . . . 22
- 1.6 Related Documentation and Resources . . . 23
- 1.7 Glossary . . . 23
- 1.8 Revision History . . . 24
- 1.9 Disclaimer and Copyright Notice . . . 24

This document provides guidelines for the ESP32-P4 SoC.

Schematic Checklist	PCB Layout Design	Hardware Development

Chapter 1

Latest Version of This Document

Check the link to make sure that you use the latest version of this document: https://docs.espressif.com/projects/esp-hardware-design-guidelines/en/latest/esp32p4/index.html

1.1 About This Document

1.1.1 Introduction

The hardware design guidelines advise on how to integrate ESP32-P4 into a product. These guidelines will help to achieve optimal performance of your product, ensuring technical accuracy and adherence to Espressif's standards. The guidelines are intended for hardware and application engineers.

1.1.2 Latest Version of This Document

Check the link to make sure that you use the latest version of this document: https://docs.espressif.com/projects/esp-hardware-design-guidelines/en/latest/esp32p4/index.html

1.2 Product Overview

ESP32-P4 is a system on a chip that integrates the following features:

- High-performance MCU with RISC-V 32-bit dual-core and single-core microprocessors
- Powerful image and voice processing capability
- 16 MB or 32 MB PSRAM in the chip's package
- 55 GPIOs, rich set of peripherals

Powered by 40 nm technology, ESP32-P4 provides a robust, highly-integrated platform, which helps meet the continuous demands for efficient power usage, compact design, security, high performance, and reliability. Typical application scenarios for ESP32-P4 include:

- Smart Home
- Industrial Automation
- Health Care
- Consumer Electronics

- Smart Agriculture
- POS Machines
- Service Robot
- Audio Devices
- Generic Low-power IoT Sensor Hubs
- Generic Low-power IoT Data Loggers
- Cameras for Video Streaming
- USB Devices
- Speech Recognition
- Image Recognition
- Touch and Proximity Sensing

Note: Unless otherwise specified, "ESP32-P4" used in this document refers to the series of chips, instead of a specific chip variant.

1.3 Schematic Checklist

1.3.1 Overview

The integrated circuitry of ESP32-P4 series chips requires only 40 electrical components (resistors, capacitors, and inductors) and a crystal, as well as an SPI flash and a direct current to direct current converter (DCDC). The high integration of ESP32-P4 allows for simple peripheral circuit design. This chapter details the schematic design of ESP32-P4.

The following figure shows a reference schematic design of ESP32-P4. It can be used as the basis of your schematic design.

Any basic ESP32-P4 circuit design may be broken down into the following major building blocks:

- *Power supply*
- *Chip power-up and reset timing*
- *Flash and PSRAM*
- *Clock source*
- *UART*
- *Strapping pins*
- *GPIO*
- *ADC*
- *SDIO*
- *USB*
- *Touch sensor*
- *Ethernet MAC*
- *MIPI*

The rest of this chapter details the specifics of circuit design for each of these sections.

1.3.2 Power Supply

The general recommendations for power supply design are:

- For a single power supply, a voltage of 3.3 V is recommended.

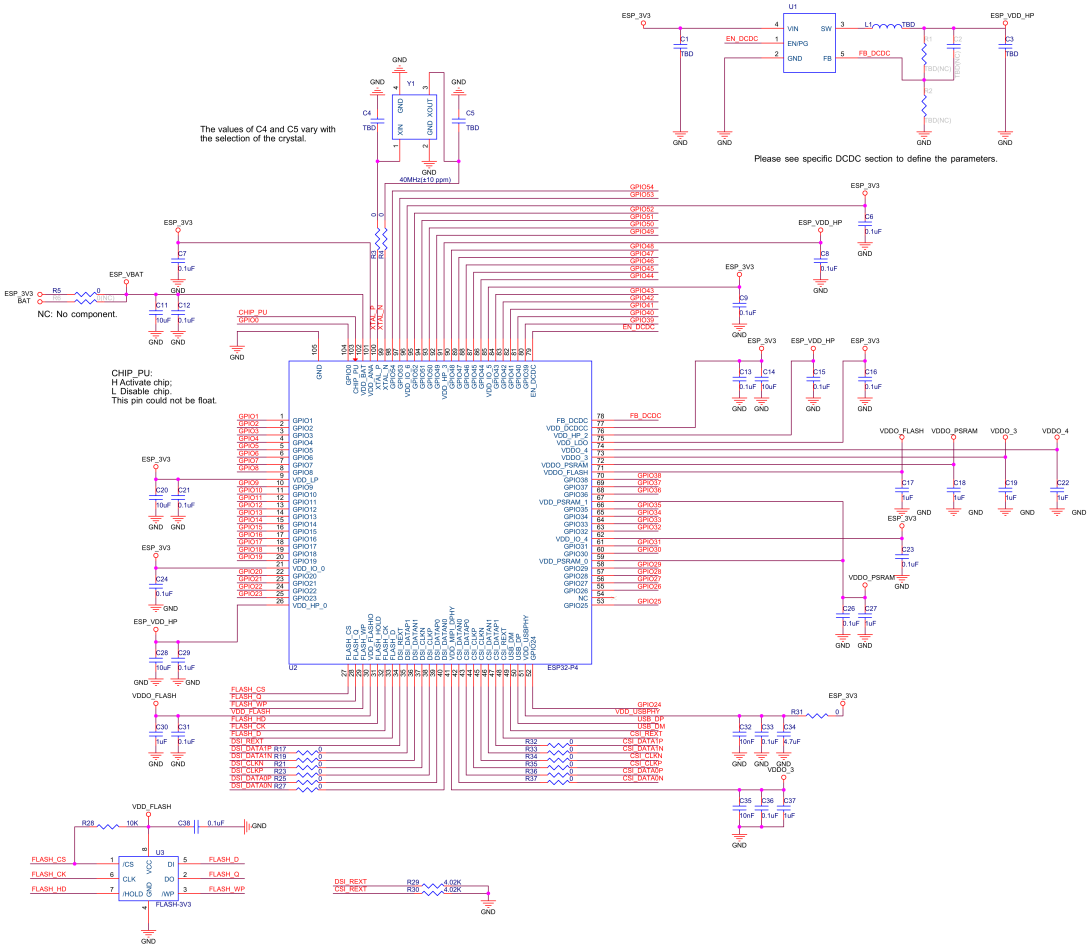

Fig. 1: ESP32-P4 Reference Schematic

- Without external peripherals, ESP32-P4 requires a minimum supply current of 430 mA. For the supply current when peripherals are connected, please refer to *HP/LP IO Power Supply*, *MIPI PHY Power Supply* and *USB PHY Power Supply*. Calculate the required supply current based on your application and choose an appropriate power supply chip.
- It is suggested to add a 10 µF capacitor at each power entrance.
- The power scheme is as shown in Figure *ESP32-P4 Power Scheme*.

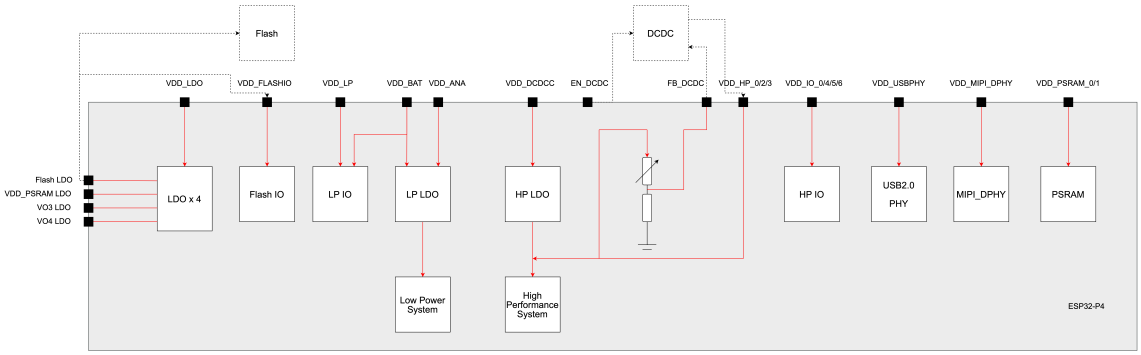

Fig. 2: ESP32-P4 Power Scheme

HP/LP IO Power Supply

Pin No.	Pin Name	Direction	Voltage (V)	Max. Current (mA)	IO Pin	Capacitors (µF)
9	VDD_LP	Input	3.0 ~ 3.6	100	LP IO	0.1
21	VDD_IO_0	Input	1.65 ~ 1.95/3.0 ~ 3.6	100	HP IO	0.1
62	VDD_IO_4	Input	1.65 ~ 1.95/3.0 ~ 3.6	100	HP IO	0.1
85	VDD_IO_5	Input	1.65 ~ 1.95/3.0 ~ 3.6	100	HP IO	0.1
96	VDD_IO_6	Input	1.65 ~ 1.95/3.0 ~ 3.6	100	HP IO	0.1

MIPI PHY Power Supply

The VDD_MIPI_DPHY of ESP32-P4 is the power supply pin for MIPI PHY. The operating voltage range is 2.25 V ~ 2.75 V. The maximum current consumption is 50 mA. It is recommended to place 10 nF + 0.1 µF + 1 µF capacitors near VDD_MIPI_DPHY in the circuit.

If the MIPI function is not required, VDD_MIPI_DPHY can be left floating.

USB PHY Power Supply

The VDD_USBPHY of ESP32-P4 is the power supply pin for USB PHY. The operating voltage range is 2.97 V ~ 3.63 V. The maximum current consumption is 20 mA. It is recommended to place 10 nF + 0.1 µF + 1 µF capacitors near VDD_USBPHY in the circuit.

If the high-speed USB 2.0 OTG function is not required, VDD_USBPHY can be left floating. When this power supply is used, it cannot be completely turned off, causing additional power consumption in low-power mode. To reduce power consumption, it is recommended to add a MOSFET circuit to fully disconnect it from the external power supply. Initially, a series resistor can be reserved on VDD_USBPHY.

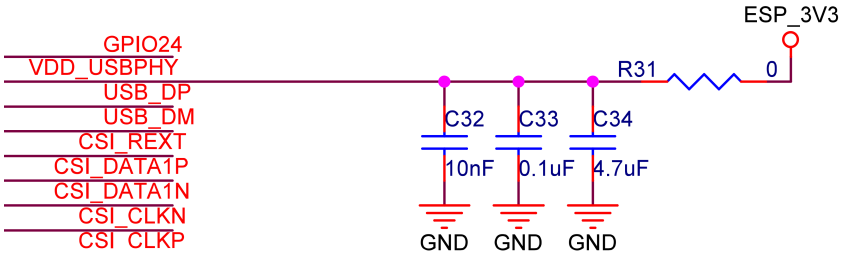

Fig. 3: VDD_USBPHY Reference Schematic

Flash and PSRAM IO Power Supply

The VDD_FLASHIO of ESP32-P4 is the power supply pin for FLASH IO. The operating voltage is 1.65 V ~ 1.95 V/3.0 V ~ 3.6 V. This power is supplied by VDDO_FLASH from the voltage regulator. It is recommended to place 0.1 µF + 1 µF capacitors near VDD_FLASHIO in the circuit.

VDD_PSRAM_0 and VDD_PSRAM_1 are power supply pins for PSRAM IO. The operating voltage is 1.65 V ~ 1.95 V. This power is supplied by VDDO_PSRAM from the voltage regulators. It is recommended to place 0.1 µF + 1 µF capacitors near VDD_PSRAM_0 and VDD_PSRAM_1 in the circuit.

Analog Power Supply

The VDD_ANA of ESP32-P4 is an analog power supply pin. The operating voltage range is 3.0 V ~ 3.6 V. It is recommended to place a 0.1 µF capacitor near VDD_ANA in the circuit. The VDD_BAT is also an analog power supply pin. The operating voltage range is 3.0 V ~ 3.6 V. It is recommended to place 0.1 µF + 10 µF capacitors near VDD_BAT.

Pin VDD_BAT must not be left floating. An external battery can be connected. For details, refer to the ESP32-P4 Battery Backup Solution.

Digital Power Supply

VDD_HP_0, VDD_HP_2, and VDD_HP_3 of the ESP32-P4 are digital power supply pins, working in a voltage range of 0.99 V ~ 1.3 V. This power is supplied by ESP_VDD_HP from the external DCDC. It is recommended to place a 10 µF capacitor at the main power source and 0.1 µF capacitors near each power pin.

Internal Voltage Regulators and External DCDC

The VDD_LDO of ESP32-P4 provides power for the low dropout regulators (LDOs). The operating voltage ranges from 3.0 V to 3.6 V. VDD_DCDCC provides power for DCDC. The operating voltage ranges from 3.0 V to 3.6 V. Due to the high current on these pins, place a 10 µF capacitor on the power traces of VDD_LDO and VDD_DCDCC, and add a 0.1 µF capacitor at each pin.

The LDO outputs VDDO_FLASH to power the external flash. The default output is 3.3 V, but it can be configured to 1.8 V by programming EFUSE_0PXA_TIEH_SEL_0.

Table 1: VDDO_FLASH Voltage

Power	EFUSE_0PXA_TIEH_SEL_0	Voltage
flash LDO	0	3.3 V
	2	1.8 V

The LDO outputs VDDO_PSRAM to power the in-package PSRAM. The typical output voltage is 1.9V, which requires software configuration. The default output is 0.

The LDO outputs VDDO_3/4 to power the peripherals. The maximum output current is 50 mA. The output voltage ranges from 0.5V to 2.7V/3.3V, which also requires software configuration. The default output is 0.

For software configuration, please refer to Low Dropout Linear Regulator (LDO).

It is recommended to place a 1 µF capacitor near the VDDO_FLASH, VDDO_PSRAM, and VDDO_3/4 in the circuit.

The VDD_HP_0/2/3 is powered by an external DC-DC converter (DCDC), with the following details:

- The input of the external DCDC is the same as the VDD_DCDCC power supply.
- EN_DCDC is the enable pin of the external DCDC. In download mode or when there is no firmware, the voltage on EN_DCDC is 0. After the firmware starts normally, EN_DCDC is controlled internally. In download mode, the EN_DCDC voltage remains at 0. In sleep mode, this pin can be used to disable the power supply to reduce power consumption.
- FB_DCDC is the feedback pin of the external DCDC. The voltage of VDD_HP_0/2/3 can be adjusted via internal software.

Please use verified DCDC models: ETA3485, SY8088, RY3420, or TLV62569. The input/output specifications and inductor parameters should follow the respective datasheets. Below is the circuit design for TLV62569.

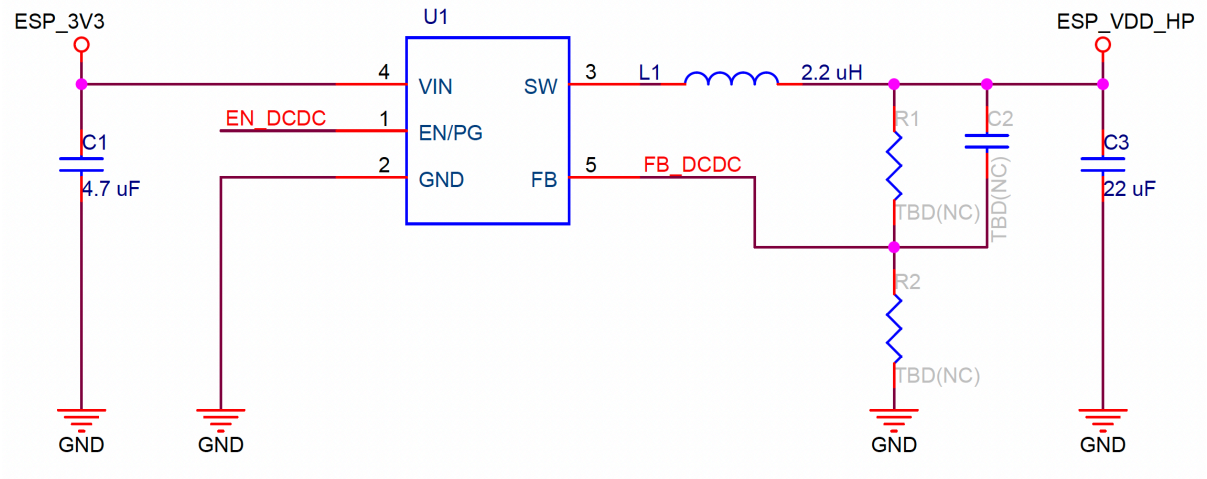

Fig. 4: TLV62569 Schematic

1.3.3 Chip Power-up and Reset Timing

ESP32-P4’ s CHIP_PU pin can enable the chip when it is high and reset the chip when it is low.

When ESP32-P4 uses a 3.3 V system power supply, the power rails need some time to stabilize before CHIP_PU is pulled up and the chip is enabled. Therefore, CHIP_PU needs to be asserted high after the 3.3 V rails have been brought up.

To reset the chip, keep the reset voltage V_{IL_nRST} in the range of (–0.3 ~ 0.25 × VDD_BAT) V. To avoid reboots caused by external interferences, make the CHIP_PU trace as short as possible.

Figure *ESP32-P4 Power-up and Reset Timing* shows the power-up and reset timing of ESP32-P4.

Table *Description of Timing Parameters for Power-up and Reset* provides the specific timing requirements.

Table 2: Description of Timing Parameters for Power-up and Reset

Parameter	Description	Minimum (µs)
t_{STBL}	Time reserved for the power rails to stabilize before the CHIP_PU pin is pulled high to activate the chip	50
t_{RST}	Time reserved for CHIP_PU to stay below V_{IL_nRST} to reset the chip	1000

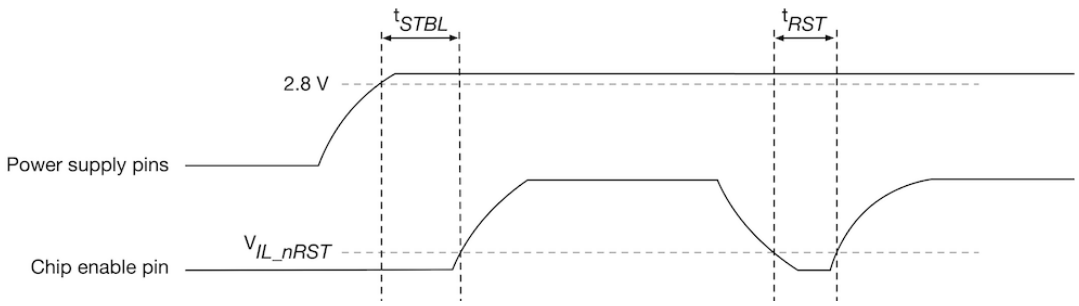

Fig. 5: ESP32-P4 Power-up and Reset Timing

Attention:

- CHIP_PU must not be left floating.
- To ensure the correct power-up and reset timing, it is advised to add an RC delay circuit at the CHIP_PU pin. The recommended setting for the RC delay circuit is usually R = 10 kΩ and C = 0.1 µF. However, specific parameters should be adjusted based on the characteristics of the actual power supply and the power-up and reset timing of the chip.
- If the user application has one of the following scenarios:
  - Slow power rise or fall, such as during battery charging.
  - Frequent power on/off operations.
  - Unstable power supply, such as in photovoltaic power generation.

  Then, the RC circuit itself may not meet the timing requirements, resulting in the chip being unable to boot correctly. In this case, additional designs need to be added, such as:
  - Adding an external reset chip or a watchdog chip, typically with a threshold of around 3.0 V.
  - Implementing reset functionality through a button or the main controller.

1.3.4 Flash and PSRAM

ESP32-P4 requires off-package flash for storing application firmware and data. ESP32-P4 supports up to 64 MB flash, which can be connected via SPI, Dual SPI, and Quad SPI.

ESP32-P4 includes OPI/HPI 1.8 V PSRAM, which is not pinned out externally.

The following table lists the pin mapping between ESP32-P4 and off-package flash for Quad SPI mode. Please note that the pins can connect to one flash at the maximum.

Table 3: Mapping of chip and package to Quad SPI Flash pins

ESP32-P4	External Package Flash (Quad SPI)
FLASH_CK	CLK
FLASH_CS	CS#
FLASH_D	DI
FLASH_Q	DO
FLASH_WP	WP#
FLASH_HOLD	HOLD#

By default, VDDO_FLASH is used as the power supply for flash, and VDDO_PSRAM is used as the power supply for PSRAM.

When VDDO_FLASH is in the output mode of 3.3 V, consider the impact of R_{VFB} on VDD_LDO. For example, when connecting a 3.3 V flash, the following condition must be met:

$$VDD_LDO > VDD_flash_min + I_flash_max \times R_{VFB}$$

In the formula above, VDD_flash_min is the minimum operating voltage of the flash, I_flash_max is the maximum operating current of the flash, and R_{VFB} is the on-resistance in 3.3 V mode.

Attention:

- It is recommended to add zero-ohm resistor footprints in series on the SPI communication lines. These footprints provide flexibility for future adjustments, such as tuning drive strength, mitigating RF interference, correcting signal timing, and reducing noise, if needed.
- Place a pull-up resistor at the FLASH_CS pin.
- It is recommended to place a 0.1 µF capacitor at the flash power supply pin.

1.3.5 Clock Source

ESP32-P4 supports two external clock sources:

- *External crystal clock source (Compulsory)*
- *RTC clock source (Optional)*

External Crystal Clock Source (Compulsory)

The ESP32-P4 firmware only supports 40 MHz crystal.

The circuit for the crystal is shown in Figure *ESP32-P4 Schematic for External Crystal*. Note that the accuracy of the selected crystal should be within ±10 ppm.

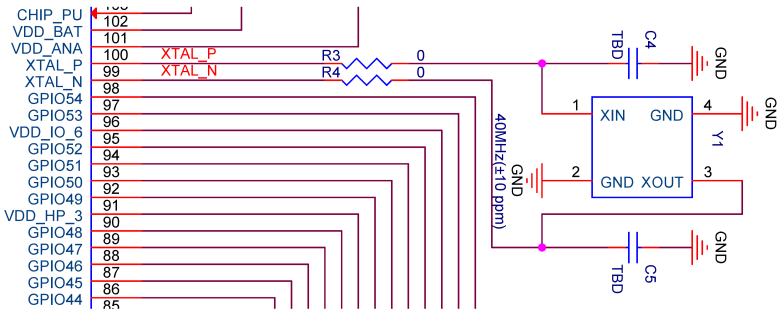

Fig. 6: ESP32-P4 Schematic for External Crystal

The initial values of external capacitors C4 and C5 can be determined according to the formula:

$$C_L = \frac{C4 \times C5}{C4 + C5} + C_{stray}$$

In the formula above, the value of C_L (load capacitance) can be found in the crystal's datasheet, and the value of C_{stray} refers to the PCB's stray capacitance. The values of C4 and C5 need to be further adjusted after an overall test.

RTC Clock Source (Optional)

ESP32-P4 supports an external 32.768 kHz crystal to act as the RTC clock. The external RTC clock source enhances timing accuracy and consequently decreases average power consumption, without impacting functionality.

Figure *ESP32-P4 Schematic for 32.768 kHz Crystal* shows the schematic for the external 32.768 kHz crystal.

Please note the requirements for the 32.768 kHz crystal:

- Equivalent series resistance (ESR) ≤ 70 kΩ.

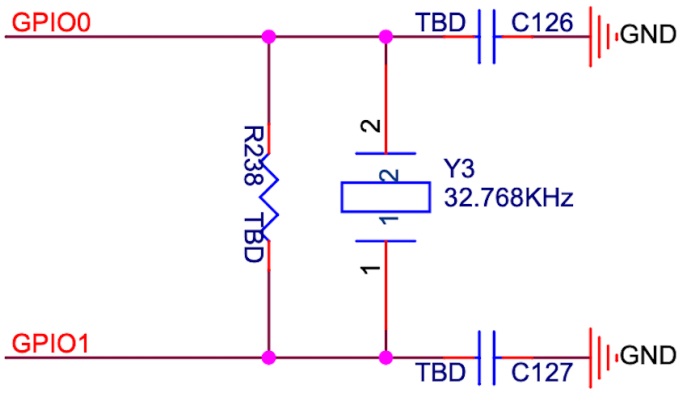

Fig. 7: ESP32-P4 Schematic for 32.768 kHz Crystal

- Load capacitance at both ends should be configured according to the crystal' s specification.

The parallel resistor R is used for biasing the crystal circuit (5 MΩ < R ≤ 10 MΩ). In general, you do not need to populate the resistor.

If the RTC clock source is not required, then the pins for the 32.768 kHz crystal can be used as GPIOs.

1.3.6 UART

ESP32-P4 includes five UART interfaces, UART0 ~ UART4, all of which support both hardware flow control (CTS and RTS signals) and software flow control (XON and XOFF). In addition, ESP32-P4 includes one LP UART interface, which also supports both hardware flow control (CTS and RTS signals) and software flow control (XON and XOFF).

The UART0_TXD pin is GPIO37, and the UART0_RXD pin is GPIO38. The other UART0 signals can be mapped to any available GPIO through software configurations. The LP UART TXD is LP GPIO14, and the LP UART RXD is LP GPIO15. The other LP UART signals can be mapped to any available GPIO through software configurations.

Typically, UART0 is used for download and log printing. For instructions on how to download via UART0, please refer to Section *Download Guidelines*.

It is recommended to use UART1 ~ UART4 as communication serial ports.

1.3.7 SPI

When using the SPI function, to improve EMC performance, add a series resistor (or ferrite bead) and a capacitor to ground on the SPI_CLK trace. If space allows, it is recommended to also add a series resistor and capacitor to ground on other SPI traces. Ensure that the RC/LC components are placed close to the pins of the chip or module.

1.3.8 Strapping Pins

At each startup or reset, a chip requires some initial configuration parameters, such as in which boot mode to load the chip, etc. These parameters are passed over via the strapping pins. After reset, the strapping pins work as normal function pins.

GPIO34, GPIO35, GPIO36, GPIO37, and GPIO38 are strapping pins.

After chip reset is released, the combination of GPIO35, GPIO36, GPIO37, and GPIO38 controls the boot mode. See Table *Chip Boot Mode Control*.

Table 4: Chip Boot Mode Control

Boot Mode	GPIO35	GPIO36	GPIO37	GPIO38
SPI Boot Mode (Default)	1	Any value	Any value	Any value
Joint Download Mode[1]	0	1	Any value	Any value

Signals applied to the strapping pins should have specific *setup time* and *hold time*. For more information, see Figure *Setup and Hold Times for Strapping Pins* and Table *Description of Timing Parameters for Strapping Pins*.

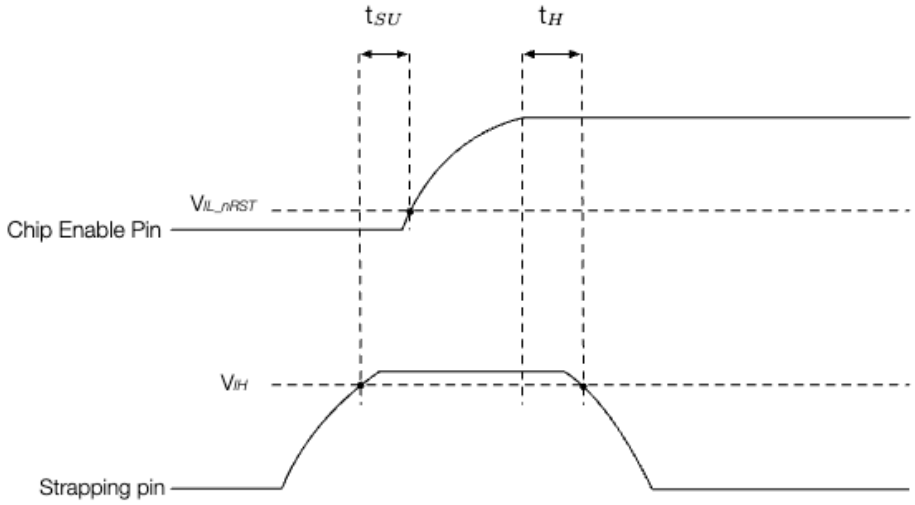

Fig. 8: Setup and Hold Times for Strapping Pins

Table 5: Description of Timing Parameters for Strapping Pins

Parameter	Description	Minimum (ms)
t_{SU}	Time reserved for the power rails to stabilize before the chip enable pin (CHIP_PU) is pulled high to activate the chip.	0
t_{H}	Time reserved for the chip to read the strapping pin values after CHIP_PU is already high and before these pins start operating as regular IO pins.	3

Attention:

- It is recommended to reserve a pull-up resistor at the GPIO35.
- Do not add high-value capacitors at GPIO35, otherwise, the chip may not boot successfully.

1.3.9 GPIO

When using GPIOs, please:

[1] The following download methods are supported under Joint Download Mode:
- USB Download Boot:
  - USB-Serial-JTAG Download Boot
  - USB 2.0 OTG Download Boot
- UART Download Boot
- SPI Slave Download Boot

- Pay attention to the states of strapping pins during power-up.
- Pay attention to the default configurations of the GPIOs after reset. The default configurations can be found in Table *IO MUX Pin Functions*. It is recommended to add a pull-up or pull-down resistor to pins in the high-impedance state or enable the pull-up and pull-down during software initialization to avoid extra power consumption.
- Only GPIOs in the VDD_LP power domain can be controlled in Deep-sleep mode.
- When powered solely by the battery, only GPIOs in the VDD_BAT power domain can be controlled.

Table 6: IO MUX Pin Functions

Pin Number	Pin Name	Power Supply Pin	At Reset	After Reset
1	GPIO1	VDD_LP/VDD_BAT	–	–
2	GPIO2	VDD_LP/VDD_BAT	–	IE, WPU
3	GPIO3	VDD_LP/VDD_BAT	–	IE
4	GPIO4	VDD_LP	–	IE
5	GPIO5	VDD_LP	–	–
6	GPIO6	VDD_LP	–	–
7	GPIO7	VDD_LP	–	–
8	GPIO8	VDD_LP	–	–
10	GPIO9	VDD_LP	–	–
11	GPIO10	VDD_LP	–	–
12	GPIO11	VDD_LP	–	–
13	GPIO12	VDD_LP	–	–
14	GPIO13	VDD_LP	–	–
15	GPIO14	VDD_LP	–	–
16	GPIO15	VDD_LP	–	–
17	GPIO16	VDD_IO_0	–	–
18	GPIO17	VDD_IO_0	–	–
19	GPIO18	VDD_IO_0	–	–
20	GPIO19	VDD_IO_0	–	–
22	GPIO20	VDD_IO_0	–	–
23	GPIO21	VDD_IO_0	–	–
24	GPIO22	VDD_IO_0	–	–
25	GPIO23	VDD_IO_0	–	–
52	GPIO24	VDD_IO_4	–	–
53	GPIO25	VDD_IO_4	–	IE, USB_PU
55	GPIO26	VDD_IO_4	–	–
56	GPIO27	VDD_IO_4	–	–
57	GPIO28	VDD_IO_4	–	–
58	GPIO29	VDD_IO_4	–	–
60	GPIO30	VDD_IO_4	–	–
61	GPIO31	VDD_IO_4	–	–
63	GPIO32	VDD_IO_4	IE	–
64	GPIO33	VDD_IO_4	IE	–
65	GPIO34	VDD_IO_4	IE	–
66	GPIO35	VDD_IO_4	IE, WPU	–
68	GPIO36	VDD_IO_4	IE	–
69	GPIO37	VDD_IO_4	IE	–
70	GPIO38	VDD_IO_4	IE	–
80	GPIO39	VDD_IO_5	–	–
81	GPIO40	VDD_IO_5	–	–
82	GPIO41	VDD_IO_5	–	–
83	GPIO42	VDD_IO_5	–	–
84	GPIO43	VDD_IO_5	–	–
86	GPIO44	VDD_IO_5	–	–
87	GPIO45	VDD_IO_5	–	–

continues on next page

Table 6 – continued from previous page

Pin Number	Pin Name	Power Supply Pin	At Reset	After Reset
88	GPIO46	VDD_IO_5	–	–
89	GPIO47	VDD_IO_5	–	–
90	GPIO48	VDD_IO_5	–	–
92	GPIO49	VDD_IO_6	–	–
93	GPIO50	VDD_IO_6	–	–
94	GPIO51	VDD_IO_6	–	–
95	GPIO52	VDD_IO_6	–	–
97	GPIO53	VDD_IO_6	–	–
98	GPIO54	VDD_IO_6	–	–
104	GPIO0	VDD_LP/VDD_BAT	–	–

1.3.10 ADC

When using the ADC function, place a 0.1 µF capacitor to ground near the pin to improve accuracy.

ADC Channel Signal	Pin Name
ADC1_CHANNEL0	GPIO16
ADC1_CHANNEL1	GPIO17
ADC1_CHANNEL2	GPIO18
ADC1_CHANNEL3	GPIO19
ADC1_CHANNEL4	GPIO20
ADC1_CHANNEL5	GPIO21
ADC1_CHANNEL6	GPIO22
ADC1_CHANNEL7	GPIO23
ADC2_CHANNEL0	GPIO49
ADC2_CHANNEL1	GPIO50
ADC2_CHANNEL2	GPIO51
ADC2_CHANNEL3	GPIO52
ADC2_CHANNEL4	GPIO53
ADC2_CHANNEL5	GPIO54

1.3.11 SDIO

ESP32-P4 series chips have only one SD/MMC host controller, which cannot operate as a slave. SDIO 2.0 and SDIO 3.0 interfaces require time-division multiplexing.

SDIO 2.0 interface signals can be configured to any available GPIO through the GPIO matrix. However, in SDIO 3.0 interface, certain signal pins are fixed and should be connected as specified in the table below.

Additionally, it is recommended to add one series pull-up resistor for the each SDIO GPIO pin and populate a capacitor to ground on the SDIO CLK line.

SDIO 3.0 protocol specifies that the voltage of SDIO GPIOs should switch automatically. Therefore, it is recommended to connect the SDIO GPIO power domain VDD_IO_5 and the ecternal pull-up power supply to VDDO_4. Additionaly, a 10 µF capacitor should be populated to VDD_IO_5. The SD card ' s power supply should be connected to the 3.3 V power supply of the system.

To reduce the power consumption of the SD card and reset it, you can populate a MOSFET at the power supply circuit of SD card and control it via GPIO.

Table 7: SDIO 3.0 Interface Configuration

Signal	Pin Name
SD1_CDATA0_PAD	GPIO39
SD1_CDATA1_PAD	GPIO40
SD1_CDATA2_PAD	GPIO41
SD1_CDATA3_PAD	GPIO42
SD1_CCLK_PAD	GPIO43
SD1_CCMD_PAD	GPIO44
SD1_CDATA4_PAD	GPIO45
SD1_CDATA5_PAD	GPIO46
SD1_CDATA6_PAD	GPIO47
SD1_CDATA7_PAD	GPIO48

1.3.12 USB

ESP32-P4 integrates a USB serial/JTAG controller that uses GPIO24 and GPIO25 as USB D- and USB D+ by default.

ESP32-P4 also features an full-speed USB On-The-Go (OTG) peripheral with an integrated transceiver, compliant with USB 2.0 specifications. By default, the full-speed USB OTG uses GPIO26 and GPIO27 as USB D- and USB D+.

When using the USB serial/JTAG controller and full-speed USB OTG, please note the following:

- The roles of the USB D- and USB D+ pins can be swapped.
- If only one function is in use, you can select either GPIO24/GPIO25 or GPIO26/GPIO27.
- If both functions are used simultaneously, GPIO24/GPIO25 and GPIO26/GPIO27 will serve as two separate pins for each function.

ESP32-P4 also features a high-speed USB OTG peripheral with an integrated transceiver, compliant with USB 2.0 specifications, where DM and DP respectively serve as USB D- and USB D+.

It is recommended to add a 22/33 Ω series resistor and optional ground capacitors on the Full-speed USB D- and USB D+ lines. These components should be placed close to the chip end.

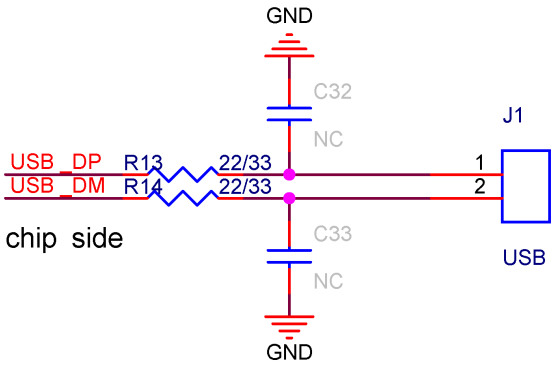

Fig. 9: ESP32-P4 USB RC Circuit

At the USB connector end, it is advisable to include ESD protection diodes. When using a high-speed USB OTG peripheral, the parasitic capacitance of the diode should not exceed 1 pF to avoid poor signal quality and unstable transmission.

ESP32-P4 also supports downloading via USB. For details, please refer to *Strapping Pins*.

1.3.13 Touch Sensor

When using the touch function, it is recommended to populate a series resistor at the chip side to reduce the coupling noise and interference on the line, and to strengthen the ESD protection. The recommended resistance is from 470 Ω to 2 kΩ, preferably 510 Ω. The specific value depends on the actual test results of the product.

The ESP32-P4 touch sensor has a waterproof design to reduce the impact of small water droplets. The ESP32-P4 touch sensor includes a special PAD called the Shield Pad. You can choose any touch pin as the Shield Pad, which will be connected in parallel with the currently measured touch pin, effectively reducing the impact of water droplets.

1.3.14 Ethernet MAC

Refer to the table below for the GPIO configuration of the EMAC RMII interface.

Table 8: EMAC Interface Configuration

Signal	Pin Name
RMII_RXDV	GPIO28, GPIO45, GPIO51
RMII_RXD0	GPIO29, GPIO46, GPIO52
RMII_RXD1	GPIO30, GPIO47, GPIO53
RMII_RXER	GPIO31, GPIO48, GPIO54
RMII_CLK	GPIO32, GPIO44, GPIO50
RMII_TXEN	GPIO33, GPIO40, GPIO49
RMII_TXD0	GPIO34, GPIO41
RMII_TXD1	GPIO35, GPIO42
RMII_TXER	GPIO36, GPIO43

It is recommended to populate a series resistor on the RMII CLK line.

Please note that the clock signal in the RMII interface is input-only. If you need a clock output solution, please use the REF_50M_CLK_PAD signal (GPIO23/GPIO39).

1.3.15 MIPI

ESP32-P4 includes an MIPI DSI interface for connecting displays with an MIPI interface, and an MIPI CSI interface for connecting cameras with an MIPI interface. Please populate a 4.02 kΩ pull-down resistor to both the CSI_REXT pin and DSI_REXT pin.

It is recommended to reserve series resistors on the MIPI communication lines (initially 0 Ω). The main purposes are to reduce drive current, adjust timing, and improve interference resistance.

If an RF module is included in the design, please also add series resistors on the CSI interface at the device end to minimize interference with the RF module.

Attention:

- If the MIPI interface is not used, the power and external resistor pins can be left floating.
- For MIPI interface devices, please use GPIO for control signals.

1.4 PCB Layout Design

This chapter introduces the key points of how to design an ESP32-P4 PCB layout using an ESP32-P4 development board as an example.

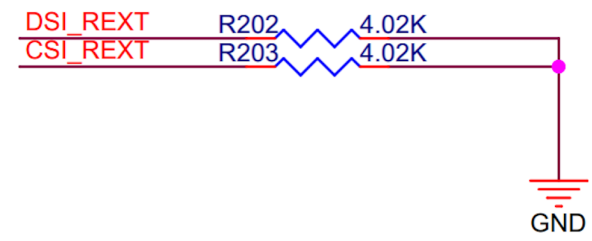

Fig. 10: ESP32-P4 Schematic for MIPI Signal

1.4.1 General Principles of PCB Layout for the Chip

Considering the communication quality of high-speed signal lines and potential interference with the RF module, please use at least a four-layer PCB design, as follows:

- Layer 1 (TOP): Signal traces and components.
- Layer 2 (GND): No signal traces here to ensure a complete GND plane.
- Layer 3 (POWER): Route power traces here. If possible, route high-speed signal traces here and ensure a complete reference plane.
- Layer 4 (BOTTOM): Route some signal traces here.

General Guidelines

- Whenever possible, route the power traces on the inner layers (not the ground layer) and connect them to the chip pins through vias. Ensure the power traces are surrounded by ground copper.
- The trace width for the 3.3 V main power supply should be at least 25 mil.
- For the power traces of VDD_LP, VDD_IO_0, VDD_FLASHIO, VDD_IO_4, VDD_IO_5 and VDD_IO_6, use a trace width of at least 10 mil. Place a 10 µF capacitor at the power entry point for this series of power supply and a 0.1 µF capacitor for each power pin.
- The trace width for the main power supply traces of VDD_HP_0, VDD_HP_2, and VDD_HP_3 should be at least 20 mil. Place a 10 µF capacitor at the power entry point for this series of power supply and a 0.1 µF capacitor for each power pin.
- For VDD_LDO and VDD_DCDCC, which handle higher current, use a trace width of at least 20 mil and place a 10 µF capacitor close to each power pin.
- It is recommended to use a star routing method to distribute power traces to each power pin.
- Place the DCDC for the VDD_HP power supply close to the chip to minimize the input, output, and feedback loops as much as possible.

1.4.2 Crystal

Figure *ESP32-P4 Crystal Layout* shows a reference PCB layout of crystal design.

The layout of the crystal should follow the guidelines below:

- Ensure a complete GND plane for the crystal and chip.
- The crystal should be placed far from the clock pin to avoid interference on the chip. The gap should be at least 4.5 mm. It is good practice to add high-density ground vias stitching around the clock trace for better isolation.
- There should be no vias for the clock input and output traces.
- Components in series to the crystal trace should be placed close to the chip side.
- The external matching capacitors should be placed on the two sides of the crystal, preferably at the end of the clock trace, but not connected directly to the series components. This is to make sure the ground pad of the capacitor is close to that of the crystal.
- Do not route high-frequency digital signal traces under the crystal. It is recommended not to route any signal trace under the crystal. The vias on the power traces on both sides of the crystal clock trace should be placed as

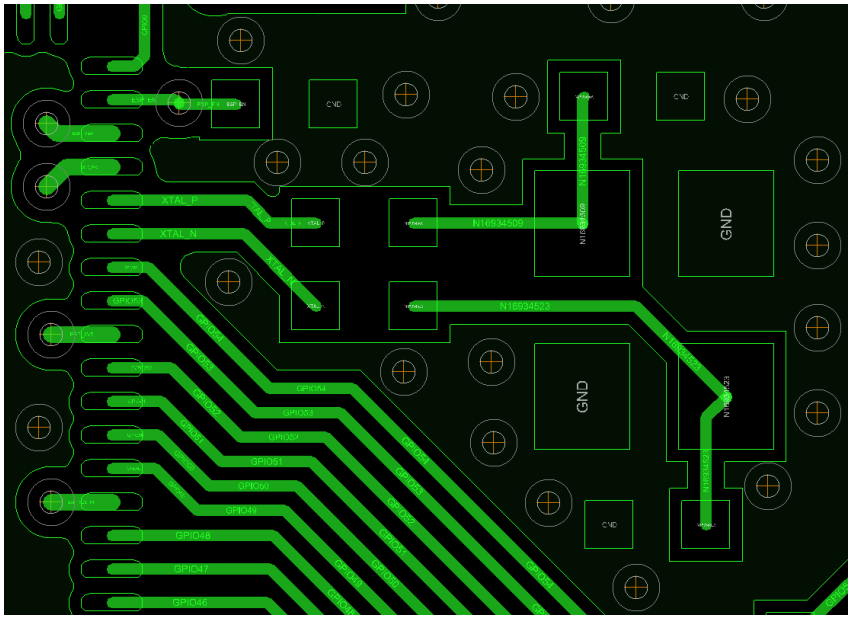

Fig. 11: ESP32-P4 Crystal Layout

far away from the clock trace as possible, and the two sides of the clock trace should be surrounded by ground copper.

- As the crystal is a sensitive component, do not place any magnetic components nearby that may cause interference, for example large inductance component, and ensure that there is a complete large-area ground plane around the crystal.

1.4.3 USB

The USB layout should meet the following guidelines:

- Reserve space for resistors and capacitors on the USB traces close to the chip side.
- For USB high speed OTG, maintain a differential pair impedance of 90 Ω with a tolerance of ±10%. Use differential pairs and route them in parallel at equal lengths.
- USB differential traces should minimize via transitions as much as possible to ensure better impedance control and avoid signal reflections. If vias are necessary, add a pair of ground return vias at each transition point.
- Ensure there is a continuous reference layer (a ground layer is recommended) beneath the USB traces.
- Surround the USB traces with ground copper.

1.4.4 SDIO

The SDIO layout should follow the guidelines below:

- Minimize parasitic capacitance of SDIO traces as they involve high-speed signals.
- The trace lengths for SDIO_CMD and SDIO_DATA0 ~ SDIO_DATA3 should be within ±50 mil of the SDIO_CLK trace length. Use serpentine routing if necessary.
- For SDIO routing, maintain a 50 Ω single-ended impedance with a tolerance of ±10%.
- Keep the total trace length from SDIO GPIOs to the master SDIO interface as short as possible, ideally within 2000 mil.
- Ensure that SDIO traces do not cross layers. Besides, a reference plane (preferably a ground plane) must be placed beneath the traces, and continuity of the reference plane must be ensured.
- It is recommended to surround the SDIO_CLK trace with ground copper.
- For multi-layer PCB designs, it is recommended to route SDIO traces to an inner layer through vias immediately after being drawn out from the chip. This helps minimize interference with high-speed signal lines. Add a pair of ground return vias at each via transition point.

1.4.5 Touch Sensor

ESP32-P4 offers up to 14 capacitive IOs that detect changes in capacitance on touch sensors due to finger contact or proximity. The chip's internal capacitance detection circuit features low noise and high sensitivity. It allows to use touch pads with smaller area to implement the touch detection function. You can also use the touch panel array to detect a larger area or more test points.

Figure *ESP32-P4 Typical Touch Sensor Application* depicts a typical touch sensor application.

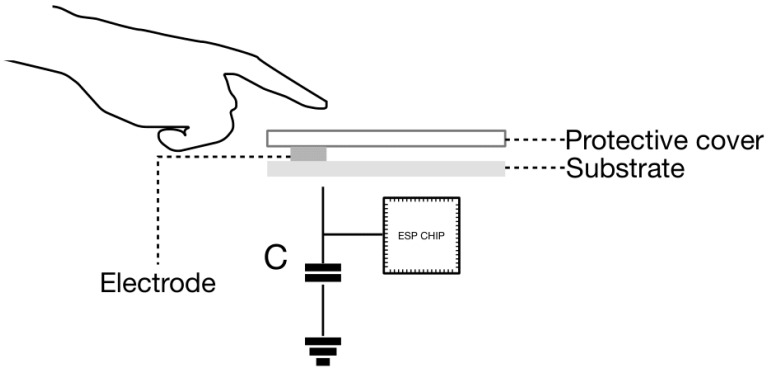

Fig. 12: ESP32-P4 Typical Touch Sensor Application

To prevent capacitive coupling and other electrical interference to the sensitivity of the touch sensor system, the following factors should be taken into account.

Electrode Pattern

The proper size and shape of an electrode improves system sensitivity. Round, oval, or shapes similar to a human fingertip are commonly applied. Large size or irregular shape might lead to incorrect responses from nearby electrodes.

Figure *ESP32-P4 Electrode Pattern Requirements* shows the proper and improper size or shape of electrode. Please note that the examples illustrated in the figure are not of actual scale. It is suggested to use a human fingertip as reference.

PCB Layout

Figure *ESP32-P4 Sensor Track Routing Requirements* illustrates the general guidelines to routing traces. Specifically,

- The trace should be as short as possible and no longer than 300 mm.
- The trace width (W) can not be larger than 0.18 mm (7 mil).
- The alignment angle (R) should not be less than 90°.
- The trace-to-ground gap (S) should be in the range of 0.5 mm to 1 mm.
- The electrode diameter (D) should be in the range of 8 mm to 15 mm.
- Hatched ground should be added around the electrodes and traces.
- The traces should be isolated well and routed away from that of the antenna.

Waterproof and Proximity Sensing Design

ESP32-P4 touch sensor has a waterproof design and features proximity sensor function. Figure *ESP32-P4 Waterproof and Proximity Sensing Design* shows an example layout of a waterproof and proximity sensing design.

Note the following guidelines to better implement the waterproof and proximity sensing design:

- The recommended width of the shield electrode width is 2 cm.

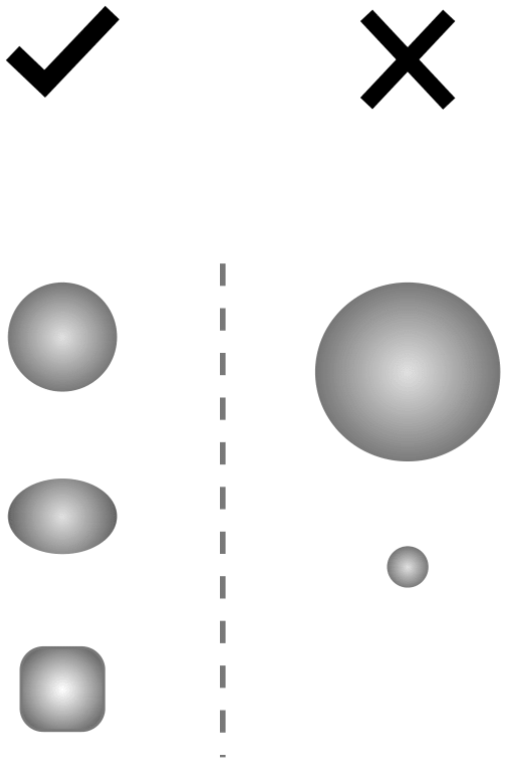

Fig. 13: ESP32-P4 Electrode Pattern Requirements

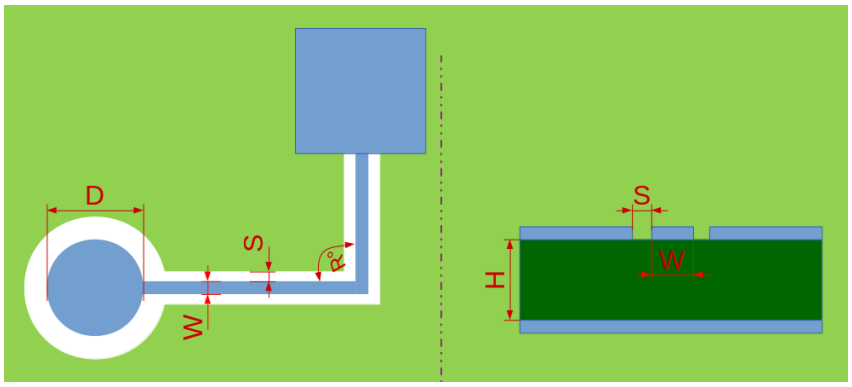

Fig. 14: ESP32-P4 Sensor Track Routing Requirements

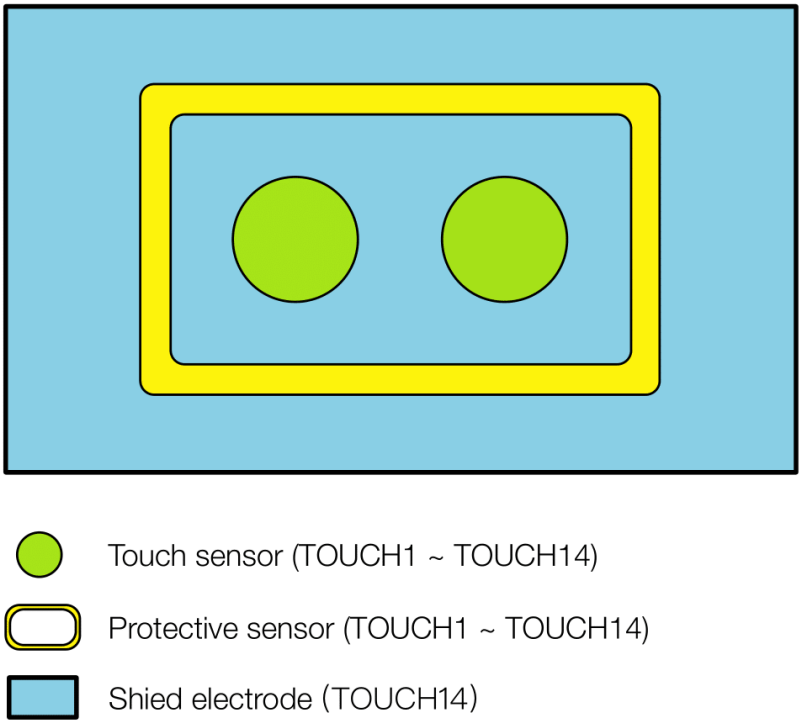

Fig. 15: ESP32-P4 Waterproof and Proximity Sensing Design

- Employ a grid on the top layer with a trace width of 7 mil and a grid width of 45 mil (25% fill). The filled grid is connected to the driver shield signal.
- Employ a grid on the bottom layer with a trace width of 7 mil and a grid width of 70 mil (17% fill). The filled grid is connected to the driver shield signal.
- The protective sensor should be in a rectangle shape with curved edges and surround all other sensors.
- The recommended width of the protective sensor is 2 mm.
- The recommended gap between the protective sensor and shield sensor is 1 mm.
- The sensing distance of the proximity sensor is directly proportional to the area of the proximity sensor. However, increasing the sensing area will introduce more noise. Actual testing is needed for optimized performance.
- It is recommended that the shape of the proximity sensor is a closed loop. The recommended width is 1.5 mm.

Note: For more details on the hardware design of the touch sensor, please refer to Touch Sensor Application Note.

1.4.6 MIPI

The MIPI layout should follow the guidelines below:

- Minimize parasitic capacitance of MIPI traces as they involve high-speed signals.
- The impedance of MIPI differential lines should be controlled to 100 Ω, with a tolerance of ±10%.
- MIPI traces should be kept equal in length and spacing. The length difference between two MIPI traces in a pair should be minimized and kept within 10 mil; the length difference between different MIPI trace pairs should be minimized and kept within 30 mil. Use serpentine routing if necessary.
- It is recommended to surround MIPI trace pairs with ground copper. If this is not feasible, maintain a minimum spacing of 2W between MIPI trace pairs, where "W" is the width of the MIPI trace. For the MIPI CLK trace, surround it with ground copper.
- MIPI signal traces should be kept away from other high-speed and high-frequency signals (such as parallel data traces and clock traces) by at least 3W and must not run parallel to them. They should also be kept away from switching power sources and other sources of interference.

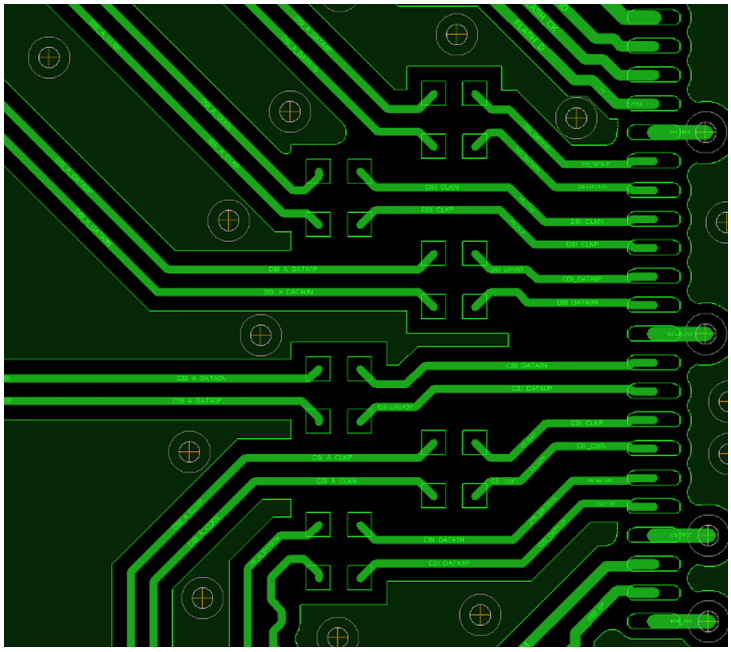

Fig. 16: ESP32-P4 MIPI Layout

- Ensure there is a continuous reference layer (preferably a ground layer) below the MIPI signal traces.
- For multi-layer PCB designs, it is recommended to route MIPI traces to an inner layer through vias immediately after being drawn out from the chip. This helps minimize interference with high-speed signal lines. Ensure the MIPI CLK traces are surrounded by ground copper. Add a pair of ground return vias at each via transition point.

1.5 Hardware Development

1.5.1 ESP32-P4 Development Boards

For a list of the latest designs of ESP32-P4 boards please check the Development Boards section on Espressif's official website.

1.5.2 Download Guidelines

You can download firmware to ESP32-P4 series chips via UART and USB.

To download via UART:

1. Before the download, make sure to set the chip to Joint Download Boot mode, according to Table *Chip Boot Mode Control*.
2. Power up the chip and check the log via the UART0 serial port. If the log shows "waiting for download", the chip has entered Joint Download Boot mode.
3. Download your firmware into flash via UART using the Flash Download Tool.
4. After the firmware has been downloaded, pull GPIO35 high or leave it floating to make sure that the chip enters SPI Boot mode.
5. Power up the chip again. The chip will read and execute the new firmware during initialization.

To download via USB:

1. If the flash is empty, set the chip to Joint Download Boot mode, according to Table *Chip Boot Mode Control*.

2. Power up the chip and check the log via USB serial port. If the log shows "waiting for download", the chip has entered Joint Download Boot mode.
3. Download your firmware into flash via USB using Flash Download Tool.
4. After the firmware has been downloaded, pull GPIO35 high or leave it floating to make sure that the chip enters SPI Boot mode.
5. Power up the chip again. The chip will read and execute the new firmware during initialization.
6. If the flash is not empty, start directly from Step 3.

Note:

- It is advised to download the firmware only after the "waiting for download" log shows via the serial port.
- Serial tools cannot be used simultaneously with the Flash Download Tool on one COM port.
- The USB auto-download will be disabled if the following conditions occur in the application, where it will be necessary to set the chip to Joint Download Boot mode first by configuring the strapping pin.
  - USB PHY is disabled by the application.
  - USB is secondary developed for other USB functions, e.g., USB host, USB standard device.
  - USB IOs are configured to other peripherals, such as UART and LEDC.
- It is recommended that the user retains control of the strapping pins to avoid the USB download function not being available in case of the above scenario.

1.6 Related Documentation and Resources

- ESP32-P4 Chip Variants
- ESP32-P4 Development Boards
- Espressif KiCad Library
- ESP Product Selector
- Regulatory Certificates
- User Forum (Hardware)
- Technical Support

1.7 Glossary

The glossary contains terms and acronyms that are used in this document.

Term	Description
CLC	Capacitor-Inductor-Capacitor
DDR SDRAM	Double Data Rate Synchronous Dynamic Random-Access Memory
ESD	Electrostatic Discharge
LC	Inductor-Capacitor
PA	Power Amplifier
RC	Resistor-Capacitor
RTC	Real-Time Clock
Zero-ohm resistor	A zero-ohm resistor acts as a placeholder in the circuit, allowing for the replacement with a higher-ohm resistor based on specific design requirements.

1.8 Revision History

This is the first release of the ESP32-P4 Hardware Design Guidelines.

1.9 Disclaimer and Copyright Notice

Information in this document, including URL references, is subject to change without notice.

All third party's information in this document is provided as is with no warranties to its authenticity and accuracy.

No warranty is provided to this document for its merchantability, non-infringement, fitness for any particular purpose, nor does any warranty otherwise arising out of any proposal, specification or sample.

All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

The Wi-Fi Alliance Member logo is a trademark of the Wi-Fi Alliance. The Bluetooth logo is a registered trademark of Bluetooth SIG.

All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners, and are hereby acknowledged.