ESP32

Hardware Design Guidelines

Version 2.6

Espressif Systems

About This Document

The guidelines outline recommended design practices when developing standalone or add-on systems based on the ESP32 series of products, including ESP32 SoCs, ESP32 modules and ESP32 development boards.

Revision History

For the revision history of this document, please refer to the last page.

Documentation Change Notification

Espressif provides email notifications to keep customers updated on changes to technical documentation. Please subscribe at www.espressif.com/en/subscribe.

Certification

Download certificates for Espressif products from www.espressif.com/en/certificates.

Disclaimer and Copyright Notice

Contents

Revision History

List of Figures

1. Overview

ESP32 is a single 2.4 GHz Wi-Fi and Bluetooth combo chip designed with TSMC ultra-low-power 40 nm technology. It is designed to achieve the best power and RF performance, robustness, versatility, and reliability in a wide variety of applications and different power profiles.

ESP32 is a highly-integrated solution for Wi-Fi + Bluetooth applications in the IoT industry with around 20 external components. ESP32 integrates the antenna switch, RF balun, power amplifier, low noise receive amplifier, filters, and power management modules. As such, the entire solution occupies minimal Printed Circuit Board (PCB) area.

ESP32 uses CMOS for single-chip fully-integrated radio and baseband, and also integrates advanced calibration circuitries that allow the solution to dynamically adjust itself to remove external circuit imperfections or adjust to changes in external conditions. As such, the mass production of ESP32 solutions does not require expensive and specialized Wi-Fi test equipment.

The ESP32 series of chips include ESP32-D0WDQ6, ESP32-D0WD, ESP32-D2WD and ESP32-S0WD. For details of part number and ordering information, please refer to *ESP32 Datasheet*.

2. Schematic Checklist and PCB Layout Design

ESP32's integrated circuitry requires only 20 resistors, capacitors and inductors, one crystal and one SPI flash memory chip. ESP32 integrates the complete transmit/receive RF functionality including the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management module, and advanced calibration circuitries.

ESP32's high integration allows for simple peripheral circuit design. This document details ESP32 schematics and PCB layout design.

While the high level of integration makes the PCB design and layout process simple, the performance of the system strongly depends on system design aspects. To achieve the best overall system performance, please follow the guidelines specified in this document for circuit design and PCB layout. All the common rules associated with good PCB design still apply and this document is not an exhaustive list of good design practices.

2.1 Schematic Checklist

ESP32 schematics is as shown in Figure 1.

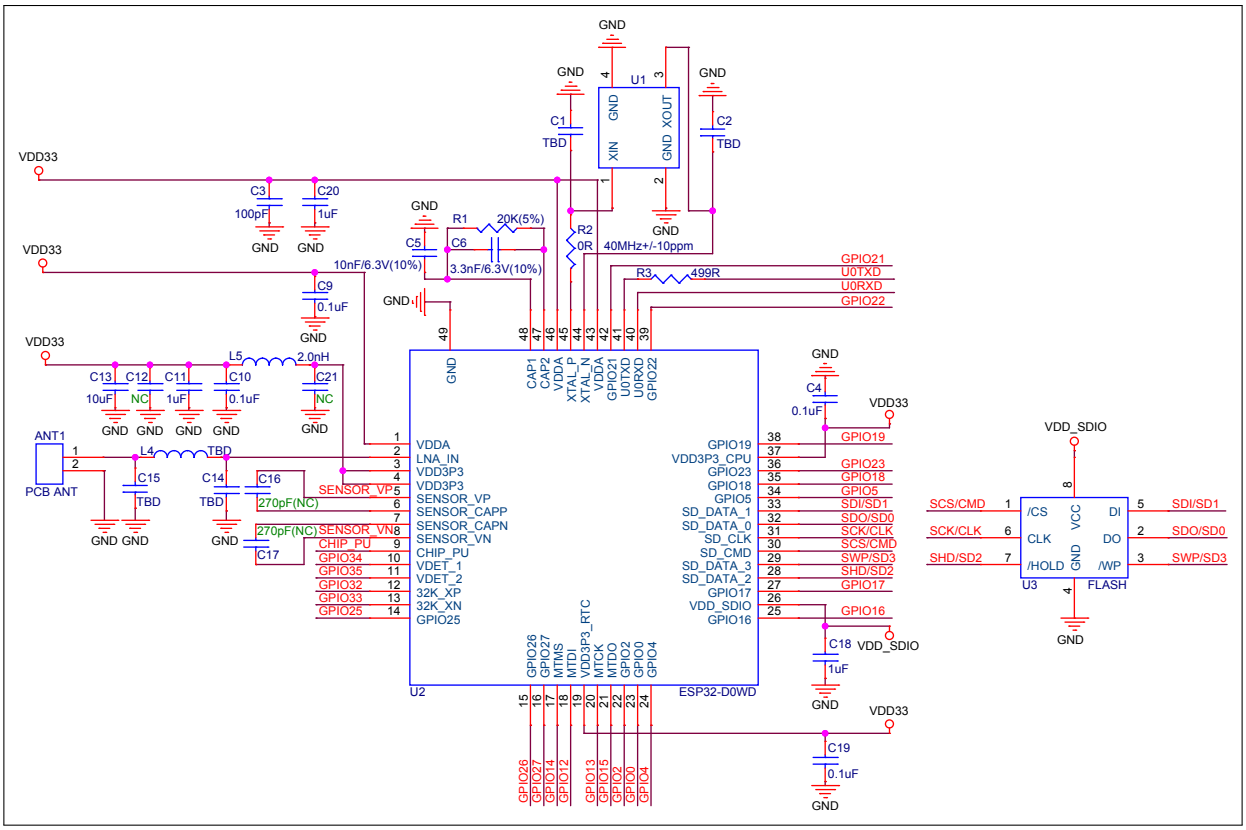

Figure 1: ESP32 Schematics (ESP32-D0WD used as an example for all illustrations in this section)

Any basic ESP32 circuit design may be broken down into eight major sections:

- Power supply
- Power-on sequence and system reset
- Flash
- Crystal oscillator

- RF
- ADC
- External capacitors
- UART

2.1.1 Power Supply

For further details of using the power supply pins, please refer to Section 2.3 Power Scheme in *ESP32 Datasheet*.

2.1.1.1 Digital Power Supply

Pin19 and pin37 are the power supply pins for RTC and CPU, respectively. The digital power supply operates in a voltage range of 1.8 V ~ 3.6 V. We recommend adding extra filter capacitors of 0.1 μF close to the digital power supply pins.

VDD_SDIO works as the power supply for the related IO, and also for an external device.

- When VDD_SDIO operates at 1.8 V, it can be generated from ESP32's internal LDO. The maximum current this LDO can offer is 40 mA, and the output voltage range is 1.65 V ~ 2.0 V. When the VDD_SDIO outputs 1.8 V, the value of GPIO12 should be set to 1 when the chip boots and it is recommended that users add a 2 kΩ ground resistor and a 4.7 μF filter capacitor close to VDD_SDIO.
- When VDD_SDIO operates at 3.3 V, it is driven directly by VDD3P3_RTC through a 6 Ω resistor, therefore, there will be some voltage drop from VDD3P3_RTC. When the VDD_SDIO outputs 3.3 V, the value of GPIO12 is 0 (default) when the chip boots and it is recommended that users add a 1 μF capacitor close to VDD_SDIO.

VDD_SDIO can also be driven by an external power supply.

When using VDD_SDIO as the power supply pin for the external 3.3 V flash/PSRAM, the supply voltage should be 2.7 V or above, so as to meet the requirements of flash/PSRAM's working voltage.

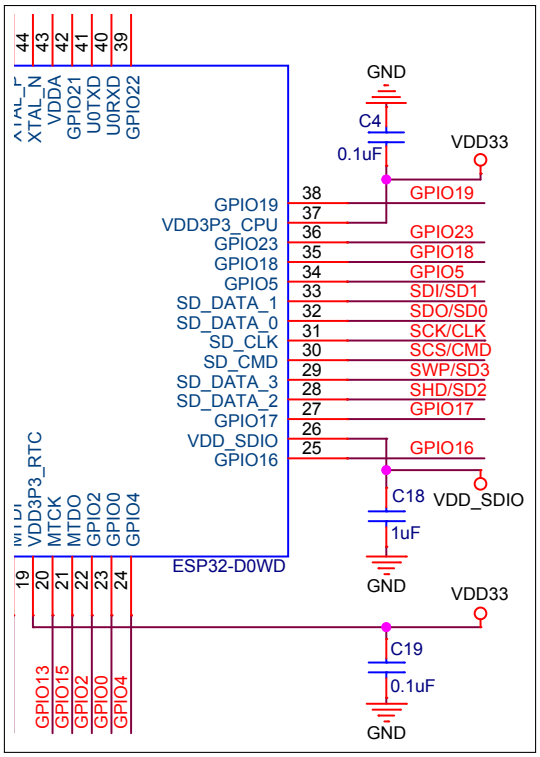

Figure 2: ESP32 Digital Power Supply Pins

2.1.1.2 Analog Power Supply

Pin1, pin3, pin4, pin43 and pin46 are the analog power supply pins. It should be noted that the sudden increase in current draw, when ESP32 is in transmission mode, may cause a power rail collapse. Therefore, it is highly recommended to add another 10 μF capacitor to the power trace, which can work in conjunction with the 0.1 μF capacitor. LC filter circuit needs to be added near the power pin so as to suppress high-frequency harmonics. The inductor's rated current is preferably 500 mA and above.

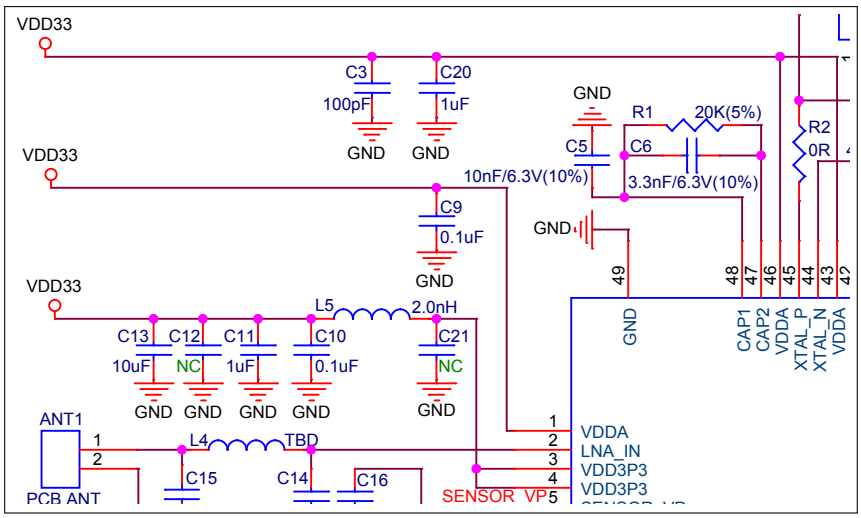

Figure 3: ESP32 Analog Power Supply Pins

Notice:

- The recommended voltage of the power supply for ESP32 is 3.3 V, and its recommended output current is 500 mA or more.
- It is suggested that users add an ESD tube at the power entrance.

2.1.2 Power-on Sequence and System Reset

2.1.2.1 Power-on Sequence

ESP32 uses a 3.3 V system power supply. The chip should be activated after the power rails have stabilized. This is achieved by delaying the activation of CHIP_PU (Pin9) after the 3.3 V rails have been brought up. More details can be found in section **Power Scheme** in *ESP32 Datasheet*.

Notice:

If CHIP_PU is driven by a power management chip, then the power management chip controls the ESP32 power state. When the power management chip turns on/off Wi-Fi through the high/low level on GPIO, a pulse current may be generated. To avoid level instability on CHIP_PU, an RC delay circuit is required.

2.1.2.2 Reset

CHIP_PU serves as the reset pin of ESP32. The input level (V_{IL_nRST}) for resetting the chip should be low enough and remain so for a period of time. More details can be found in section **Power Scheme** in *ESP32 Datasheet*.

To avoid reboots caused by external interferences, the CHIP_PU trace should be as short as possible and routed away from the clock lines. A pull-up resistor and a ground capacitor are highly recommended.

Notice:

CHIP_PU pin must not be left floating.

2.1.3 Flash

ESP32 can support up to four 16 MB external QSPI flash and SRAM chips. The demo flash used currently is an SPI flash with 4 MB ROM, in an SOP8 (208 mil) package. The VDD_SDIO acts as the power supply pin. Make sure to select the appropriate flash according to the power voltage on VDD_SDIO.

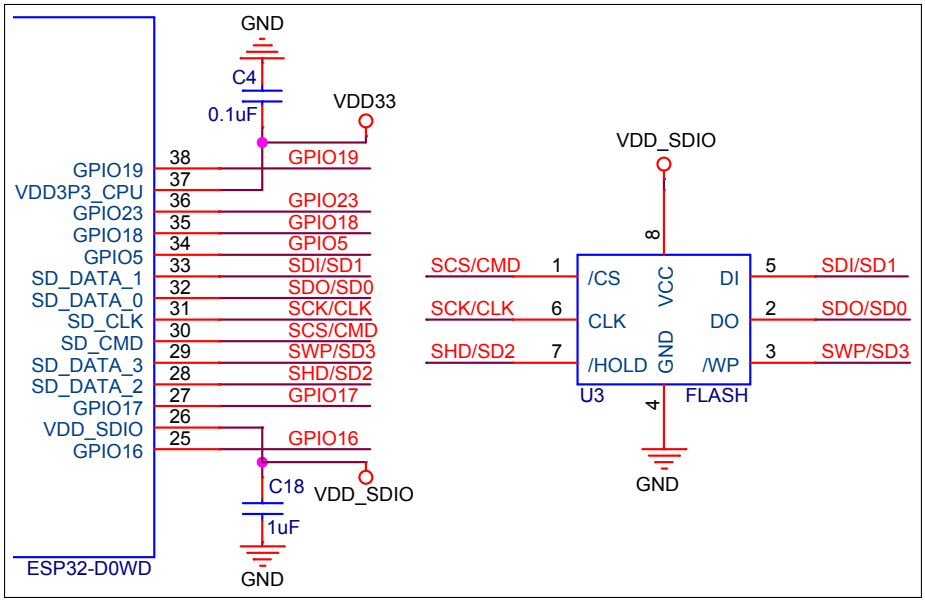

Figure 4: ESP32 Flash

2.1.4 Crystal Oscillator

There are two clock sources for the ESP32, that is, an external crystal oscillator clock source and an RTC clock source.

2.1.4.1 External Clock Source (Compulsory)

Currently, the ESP32 Wi-Fi/BT firmware only supports 40 MHz crystal oscillator. In circuit design, capacitors C1 and C2 which connect to the ground are added to the input and output terminals of the crystal oscillator respectively. The specific capacitive values depend on further testing of, and adjustment to, the overall performance of the whole circuit. It is recommended that users reserve a series resistor of 0 Ω on the XTAL_P clock trace to reduce the drive strength of the crystal, as well as to minimize the impact of crystal harmonics on RF performance. Note that the accuracy of the selected crystal is ±10 ppm.

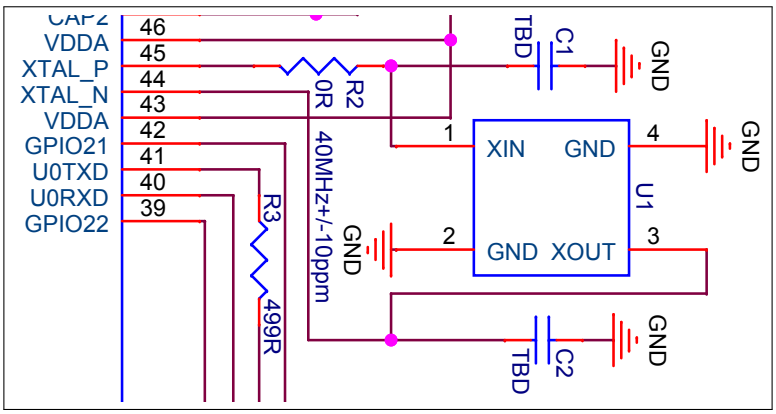

Figure 5: ESP32 Crystal Oscillator

Notice:

- If an oscillator is used, its output should be connected to XTAL_P on the chip. Please make sure that the oscillator output is stable and its accuracy is within ± 10 ppm. It is also recommended that the circuit design for the oscillator is compatible with the use of crystal, in case that if there is a defect in the circuit design, users can still use the crystal.
- Defects in the craftsmanship of the crystal oscillators (for example, frequency deviation more than ±10 ppm) and unstable operating temperature may lead to the malfunction of ESP32, resulting in a decrease of the overall performance.

2.1.4.2 RTC (Optional)

ESP32 supports an external 32.768 kHz crystal or an external 32.768 kHz signal (e.g., an oscillator) to act as the RTC sleep clock.

Figure 6 shows the schematic of the external 32.768 kHz crystal.

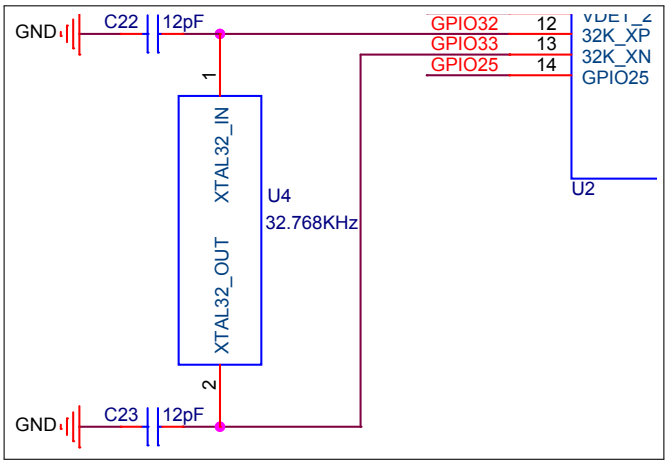

Figure 6: Schematic of ESP32 Crystal (RTC)

Notice: If the RTC source is not required, then pin12 (32K_XP) and pin13 (32K_XN) can be used as GPIOs.

Figure 7 shows the schematic of the external signal.

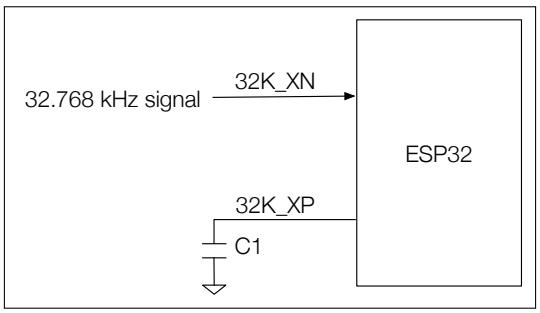

Figure 7: Schematic of External Oscillator

The value of C1 should be larger than 200 pF. The signal should meet the following requirements:

32K_XN input	Amplitude (Vpp, unit: V)
Sine wave or square wave	0.6 < Vpp < VDD

2.1.5 RF

The output impedance of the RF pins of ESP32 (QFN 6*6) and ESP32 (QFN 5*5) are (30+j10) Ω and (35+j10) Ω, respectively. A π-type matching network is essential for antenna matching in the circuit design. CLC structure is recommended for the matching network.

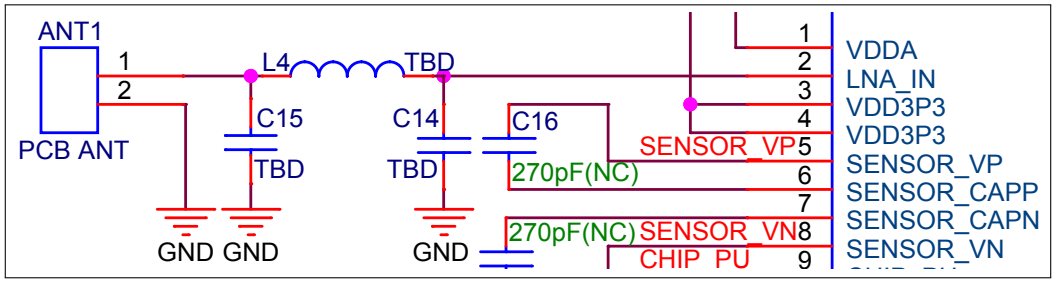

Figure 8: ESP32 RF Matching Schematics

Note:
The parameters of the components in the matching network are subject to the actual antenna and PCB layout.

2.1.6 ADC

It is recommended that users add a 0.1 μF filter capacitor to a pad when using the ADC function.

- Pins SENSOR_VP or SENSOR_VN will trigger an input glitch lasting for 80 ns once SARADC1, or SARADC2, or Hall sensor is initialized.
- Pins SENSOR_VP or SENSOR_VN is recommended for use as ADC.
- If SENSOR_VP and SENSOR_VN are used as GPIOs, while ADC is supported by other pins in the circuit design, users need to do settings in software to avoid the input glitch.

2.1.7 External Capacitor

Figure 9 shows the schematic of components connected to pin47 CAP2 and pin48 CAP1. C5 (10 nF) that connects to CAP1 should be of 10% tolerance and is required for proper operation of ESP32. RC circuit between CAP1 and CAP2 pins may be omitted under certain conditions. This circuit is used when entering Deep-sleep mode.

During this process, to minimize power consumption, the voltage to power ESP32 internals is dropped from 1.1 V to around 0.7 V. The RC circuit is used to minimize the period of the voltage drop. If removed, this process will take longer and the power consumption in Deep-sleep will be higher. If particular application of ESP32 is not using Deep-sleep mode, or power consumption is less critical, then this circuit is not required.

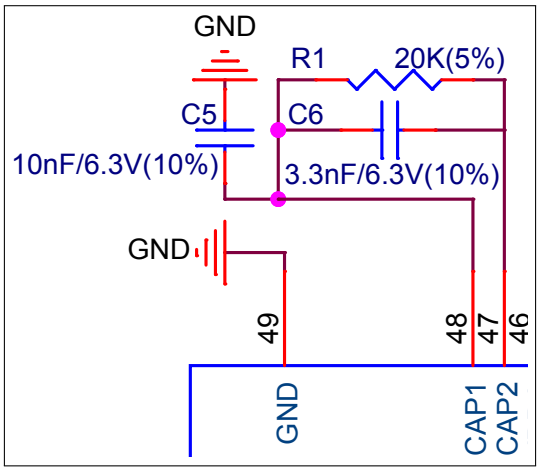

Figure 9: ESP32 External Capacitor

2.1.8 UART

Users need to connect a 499 Ω resistor to the U0TXD line in order to suppress the 80 MHz harmonics.

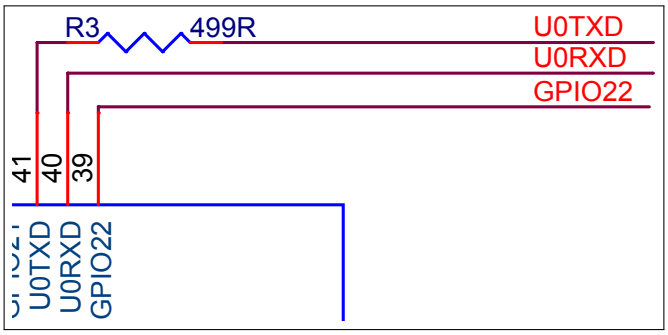

Figure 10: ESP32 UART

2.2 PCB Layout Design

This chapter introduces the key points of designing ESP32 PCB layout with the example of ESP32-WROOM-32D.

The PCB layout design guidelines are applicable to cases when the

- ESP32 module functions as a standalone device, and when the
- ESP32 functions as a slave device.

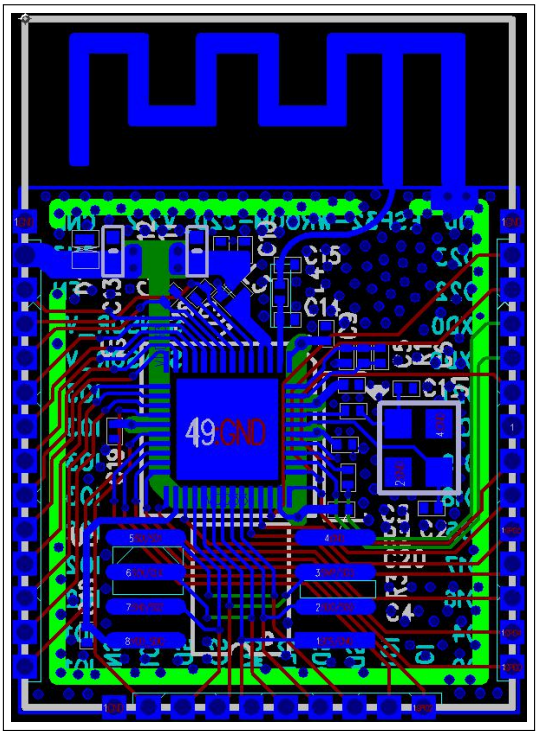

Figure 11: ESP32 PCB Layout

2.2.1 Standalone ESP32 Module

2.2.1.1 General Principles of PCB Layout

We recommend a four-layer PCB design.

- The first layer is the TOP layer for signal traces and components.
- The second layer is the GND layer without signal traces being routed so as to ensure a complete GND plane.
- The third layer is the POWER layer. It is acceptable to route signal traces on this layer, provided that there is a complete GND plane under the RF and crystal oscillator.
- The fourth layer is the BOTTOM layer, where power traces are routed. Placing any components on this layer is not recommended.

Below are the suggestions for a two-layer PCB design.

- The first layer is the TOP layer for traces and components.
- The second layer is the BOTTOM layer. Please do not place any components on this layer and keep traces to a minimum. Ideally, it should be a complete GND plane.

2.2.1.2 Positioning a ESP32 Module on a Base Board

If users adopt on-board design, they should pay attention to the layout of the module on the base board. The interference of the base board on the module's antenna performance should be reduced as much as possible.

It is recommended that the PCB antenna area of the module be placed outside the base board while the module be put as close as possible to the edge of the base board so that the feed point of the antenna is closest to the board.

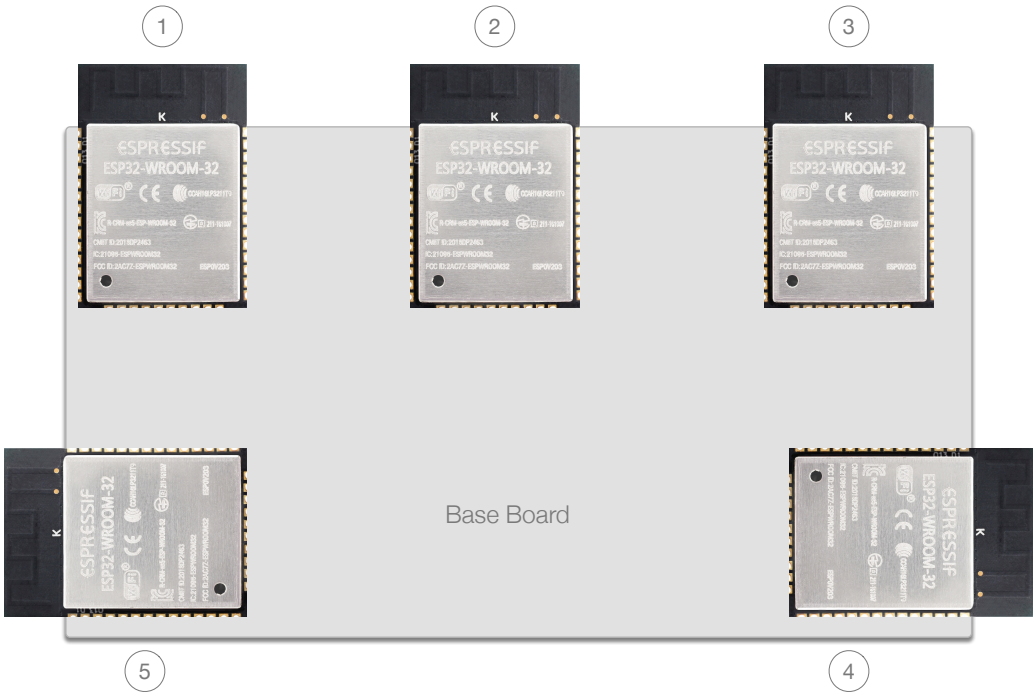

Figure 12: ESP32 Module Antenna Position on Base Board

Note:

As is shown in Figure 12, the recommended position of ESP32 module on the base board should be:

- Position 3: Highly recommended;
- Position 4: Recommended;
- Position 1, 2, 5: Not recommended.

If the positions recommended are not suitable, please make sure that the module is not covered by any metal shell. The antenna area of the module and the area 15 mm outside the antenna should be kept clean, (namely no copper, routing, components on it) as shown in Figure 13:

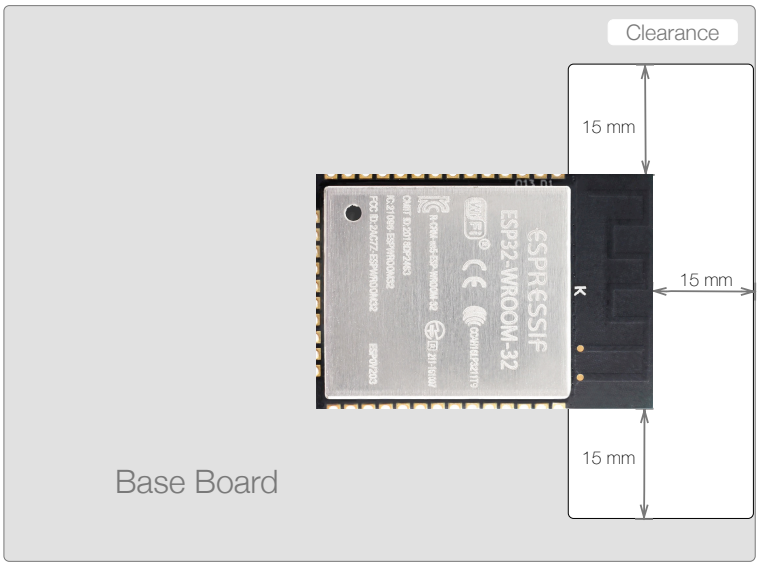

Figure 13: Keepout Zone for ESP32 Module's Antenna on the Base Board

2.2.1.3 Power Supply

- **Four-layer PCB design**

  In a four-layer PCB design, the 3.3 V power traces are routed as shown highlighted in yellow in Figure 14. The width of these power traces should be greater than 25 mil. Before power traces reach the analog power-supply pins (pin 1, 3, 4, 43, 46), a 10 μF capacitor is required, which can work in conjunction with the 0.1 μF capacitor. As Figure 14 shows, C13 (10 μF capacitor) is placed by the 3.3 V stamp hole; C10, L5 and C21 are placed as close as possible to the analog power-supply pin. If possible, add a 0.1 μF capacitor for every digital power pin. Note that all decoupling capacitors should be placed close to the power pin, and ground vias should be added adjacent to the ground pin for the decoupling capacitors to ensure a short return path.

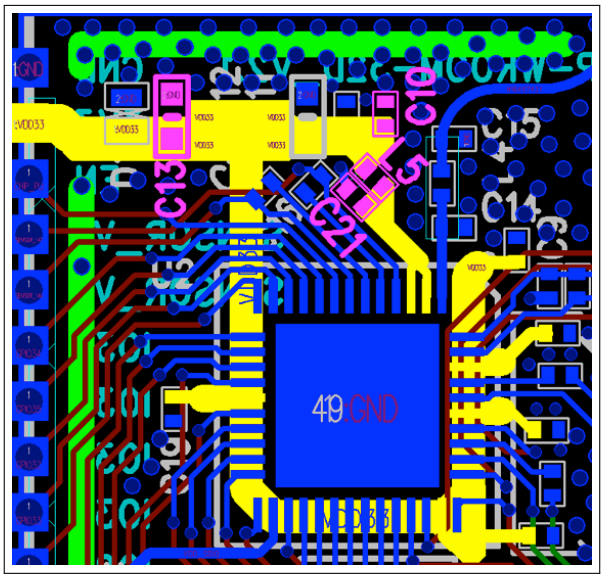

Figure 14: ESP32 Power Traces in a Four-layer PCB Design

  It is good practice to route the power traces on the fourth (bottom) layer. Vias are required for the power traces to go through the layers and get connected to the pins on the top layer. The diameter of the drill should exceed the width of the power traces. The diameter of the via pad should be 1.5 times that of the drill.

  The central thermal pad at the bottom of the chip should be connected to the ground plane through at least nine ground vias.

- **Two-layer PCB design**

  In a two-layer PCB design, the 3.3 V power traces are routed as shown labelled with VDD33 in Figure 15. In contrast to the design practices for a four-layer PCB design, the power traces in a two-layer PCB design should be routed on the top layer, thus requiring a reduced size of a thermal pad in the center of the chip. Route the power traces between the thermal pad and its surrounding signal pins. Employ vias only when the power traces have to reach the bottom layer. The purpose of this practice is to maintain a complete ground plane while reducing the surrounding area of the power traces.

  Other good practices for routing power traces in four-layer PCB designs still apply to two-layer PCB designs.

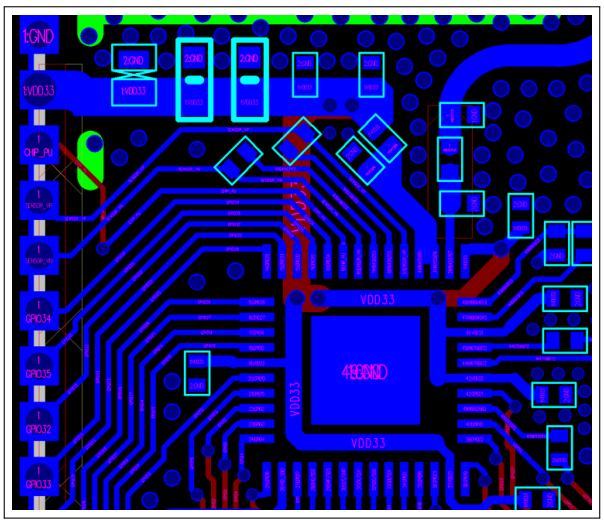

Figure 15: ESP32 Power Traces in a Two-layer PCB Design

2.2.1.4 Crystal Oscillator

For design of the crystal oscillator section, please refer to Figure 16. In addition, the following should be noted:

- The crystal oscillator should be placed far from the clock pin. **The recommended gap is 2.7 mm**. It is good practice to add high-density ground via stitching around the clock trace for containing the high-frequency clock signal.
- There should be no vias for the clock input and output traces, which means the traces cannot cross layers.
- The external regulating capacitor should be placed on the near left or right side of the crystal oscillator and at the end of the clock trace.
- Do not route high-frequency digital signal traces under the crystal oscillator. It is best not to route any signal trace under the crystal oscillator. The larger the copper area on the top layer is, the better.
- As the crystal oscillator is a sensitive component, do not place any magnetic components nearby that may cause interference, for example, power-switching converter components or unshielded inductors.

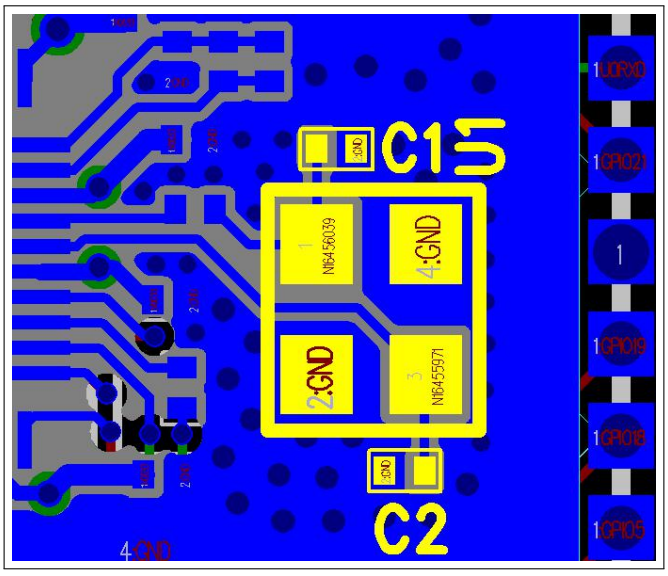

Figure 16: ESP32 Crystal Oscillator Layout

2.2.1.5 RF

- **Four-layer PCB design**

  In a four-layer PCB design, the RF trace is routed as shown highlighted in pink in Figure 17. The characteristic RF impedance must be 50 Ω. The ground plane on the adjacent layer needs to be complete. Make sure to keep the width of the RF trace consistent, and do not branch the trace. The RF trace should be as short as possible with dense ground via stitching around it for isolation.

  However, there should be no vias for the RF trace. The RF trace should be routed at a 135° angle, or with circular arcs if trace bends are required.

  π-type matching circuitry should be reserved on the RF trace and placed close to the chip.

  No high-frequency signal traces should be routed close to the RF trace. The RF antenna should be placed away from high-frequency transmitting devices, such as crystal oscillators, DDR, and clocks (SDIO_CLK), etc.

  In addition, the USB port, USB to UART chip, UART signal lines (including traces, vias, test points, header pins, etc.) must be as far away from the antenna as possible. It is good practice to add ground vias around the UART signal line. It is recommended that the design of PCB onboard antenna be based on Espressif's Type-B antenna.

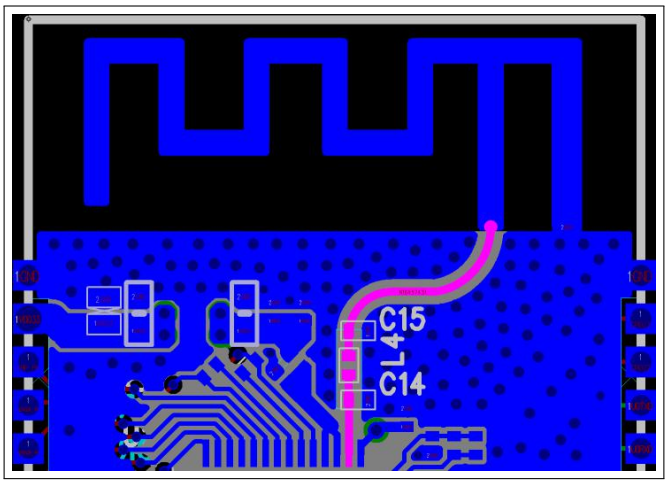

Figure 17: ESP32 RF Layout in a Four-layer PCB Design

- **Two-layer PCB design**

  In a two-layer PCB design, the RF trace is routed as shown highlighted in pink in Figure 18. The width of the RF trace should be greater than that of the RF trace in a four-layer board and is normally over 20 mil. The actual width depends on the impedance formula where impedance-relevant parameters may vary depending on the number of PCB layers.

  Other good practices for routing RF traces in four-layer PCB designs still apply to two-layer board designs.

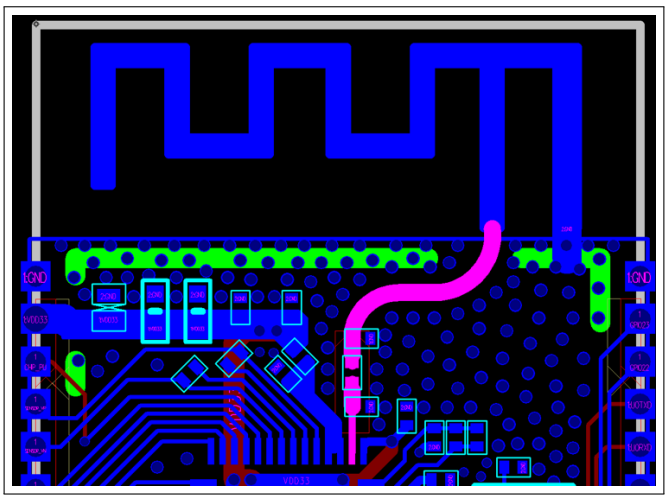

Figure 18: ESP32 RF Layout in a Two-layer PCB Design

2.2.1.6 External RC

External resistors and capacitors should be placed close to the chip pins, and there should be no vias around the traces. Please ensure that 10 nF capacitors are placed close to the pins.

2.2.1.7 UART

The series resistor on the U0TXD line needs to be placed as close to the chip as possible. The U0TXD and U0RXD traces on the top layer should be as short as possible. Employ vias around the traces for isolation.

Figure 19 below shows an example of UART design.

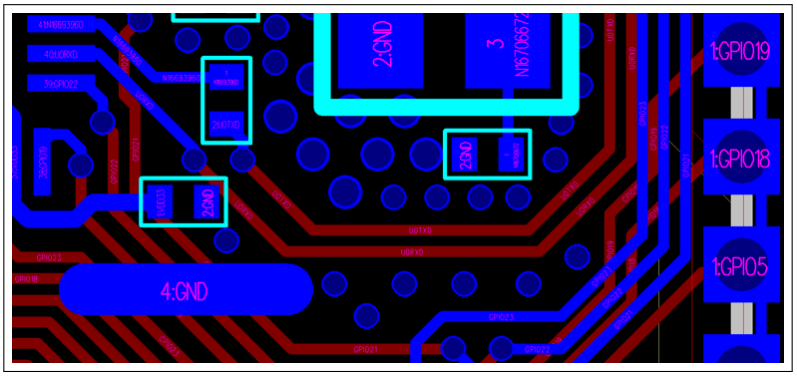

Figure 19: ESP32 UART Design

2.2.1.8 Touch Sensor

ESP32 offers up to 10 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 users to use touch pads with smaller area to implement the touch detection function. Users can also use the touch panel array to detect a larger area or more test points. Figure 20 depicts a typical touch sensor application.

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

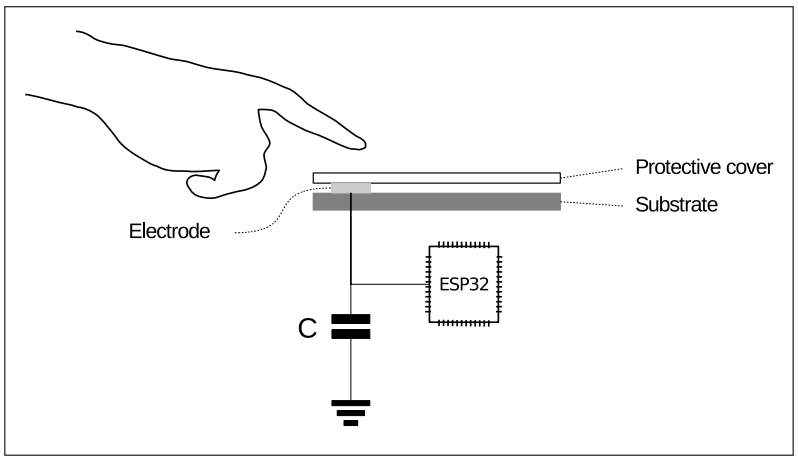

Figure 20: A Typical Touch Sensor Application

Electrode Pattern

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

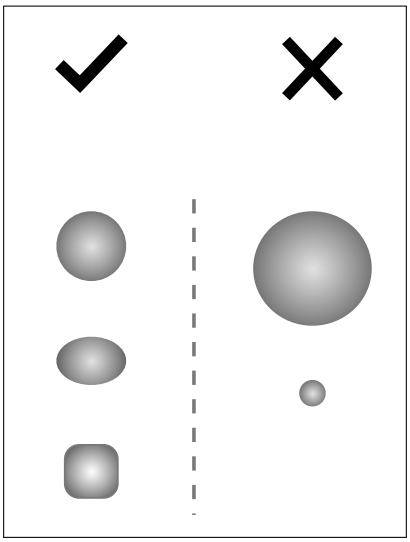

Figure 21: Electrode Pattern Requirements

Note:
The examples illustrated in Figure 21 are not of actual scale. It is suggested that users use a human fingertip as reference.

PCB Layout

The following are general guidelines to routing traces:

- The trace length should not exceed 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.

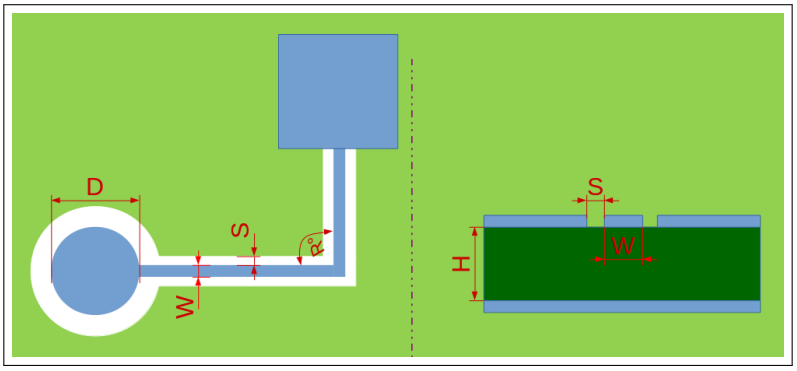

Figure 22: Sensor Track Routing Requirements

Note:
For more details on the hardware design of ESP32 touch sensor, please refer to *ESP32 Touch Sensor Application Note*.

2.2.2 ESP32 as a Slave Device

When ESP32 works as a slave device in a system, the user needs to pay more attention to signal integrity in the PCB design. It is important to keep ESP32 away from the interferences caused by the complexity of the system and an increased number of high-frequency signals. We use the mainboard of a PAD or TV Box as an example here to provide guidelines for the PCB layout and design.

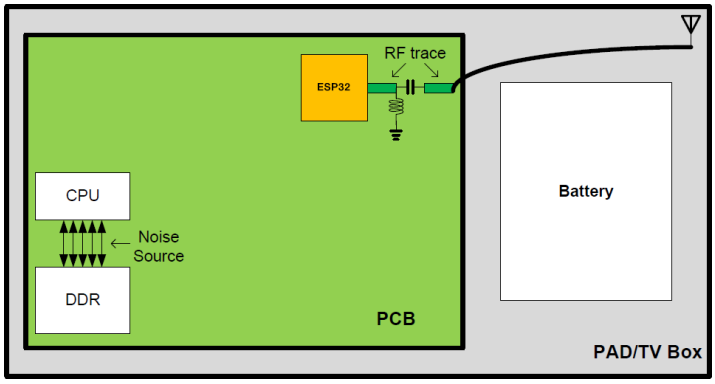

Figure 23: PAD/TV Box Layout

The digital signals between the CPU and DDR are the main producers of the high-frequency noise that interferes with Wi-Fi radio. Therefore, the following should be noted with regards to the PCB design.

- As can be seen in Figure 23, ESP32 should be placed near the edge of the PCB and away from the CPU and DDR, the main high-frequency noise sources. The distance between the chip and the noise sources decreases the interference and reduces the coupled noise.
- It is suggested that a series resistor be reserved on the six signal traces when ESP32 communicates with the CPU via SDIO to decrease the drive current and any interference, and also to eliminate the sequencing problem caused by the inconsistent length of the SDIO traces.
- On-board PCB antenna is not recommended, as it receives much interference and coupling noise, both of which impact the RF performance. We suggest that you use an external antenna which should be directed away from the PCB board via a cable, in order to weaken the high frequency interference with Wi-Fi.

- The high-frequency signal traces between the CPU and associated memory should be routed strictly according to the routing guidelines (please refer to the DDR trace routing guidelines). We recommend that you add ground vias around the CLK traces separately, and around the parallel data or address buses.
- The GND of the Wi-Fi circuit and that of other high-power devices should be separated and connected through wires if there are high-power components, such as motors, in the design.
- The antenna should be kept away from high-frequency noise sources, such as LCD, HDMI, Camera Sensor, USB, etc.

2.2.3 Typical Layout Problems and Solutions

2.2.3.1 Q: The current ripple is not large, but the TX performance of RF is rather poor.

Analysis:

The current ripple has a strong impact on the RF TX performance. It should be noted that the ripple must be tested when ESP32 is in the normal working mode. The ripple increases when the power gets high in a different mode.

Generally, the peak-to-peak value of the ripple should be <80 mV when ESP32 sends MCS7@11n packets, and <120 mV when ESP32 sends 11b/11m packets.

Solution:

Add a 10 μF filter capacitor to the branch of the power trace (the branch powering the ESP32 analog power pin). The 10 μF capacitor should be as close to the analog power pin as possible for small and stable current ripples.

2.2.3.2 Q: The power ripple is small, but RF TX performance is poor.

Analysis:

The RF TX performance can be affected not only by power ripples, but also by the crystal oscillator itself. Poor quality and big frequency offsets of the crystal oscillator decrease the RF TX performance. The crystal oscillator clock may be corrupted by other interfering signals, such as high-speed output or input signals. In addition, high-frequency signal traces, such as the SDIO trace and UART trace under the crystal oscillator, could also result in the malfunction of the crystal oscillator. Besides, sensitive components or radiation components, such as inductors and antennas, may also decrease the RF performance.

Solution:

This problem is caused by improper layout and can be solved by re-layout. Please see section 2.2 for details.

2.2.3.3 Q: When ESP32 sends data packages, the power value is much higher or lower than the target power value, and the EVM is relatively poor.

Analysis:

The disparity between the tested value and the target value may be due to signal reflection caused by the impedance mismatch on the transmission line connecting the RF pin and the antenna. Besides, the impedance mismatch will affect the working state of the internal PA, making the PA prematurely access the saturated region in an abnormal way. The EVM becomes poor as the signal distortion happens.

Solution:

Match the antenna's impedance with the reserved π-type circuit on the RF trace, so that impedance of the antenna as seen from the RF pin matches closely with that of the chip. This reduces reflections to the minimum.

2.2.3.4 Q: TX performance is not bad, but the RX sensitivity is low.

Analysis:

Good TX performance indicates proper RF impedance matching. External coupling to the antenna can affect the RX performance. For instance, the crystal oscillator signal harmonics could couple to the antenna. If the TX and RX traces of UART cross over with RF trace, then, they will affect the RX performance, as well. If ESP32 serves as a slave device, there will be other high-frequency interference sources on the board, which may affect the Rx performance.

Solution:

Keep the antenna away from crystal oscillators. Do not route high-frequency signal traces close to the RF trace. High performance digital circuitry should be placed away from the RF block on large board designs. Please see section 2.2 for details.

3. Hardware Development

Espressif designs and manufactures a large variety of modules and boards to help users evaluate functionality of the ESP32 family of chips. For a list of the latest versions of ESP32 modules and development boards, please refer to document *ESP32 Modules and Boards*.

Notes on Using Modules

- The module uses one single pin as the power supply pin. Users can connect the module to a 3.3 V power supply. The 3.3 V power supply works both for the analog circuit and the digital circuit.
- The EN pin is used for enabling the chip. Set the EN pin high for normal working mode. There is no RC delay circuit on the module. It is recommended that users add an external RC delay circuit to the module. For details please refer to Section 2.1.2.
- Lead the GND, RXD, TXD pins out and connect them to a USB-to-TTL tool for firmware download, log-printing and communication.

By default, the initial firmware has already been downloaded in the flash. If users need to re-download the firmware, they should follow the steps below:

1. Set the module to UART Download mode by pulling IO0 (pulled up by default) and IO2 (pulled down by default) low. The chip IOs are pulled down internally by default.
2. Power on the module and check through the serial terminal if the UART Download mode is enabled.
3. Download the firmware to flash, using the Flash Download Tool.
4. After downloading, pull IO0 high or just leave it floating and use the internal weak pull-up to enable the SPI Boot mode.
5. Power on the module again. The chip will read and execute the firmware during initialization.

Notice:

- During the whole process, users can check the status of the chip with the log printed through UART. If the firmware cannot be downloaded or executed, users can check if the working mode is normal during the chip initialization by looking at the log.
- The serial tool cannot be used for both the log-print and flash-download tools simultaneously.

4. Applications

4.1 ESP32 Smart Audio Platform

4.1.1 ESP32-LyraT Audio Development Board

ESP32-LyraT is an open-source development board for Espressif Systems' Audio Development Framework, ESP-ADF. It is designed for smart speakers and smart-home applications. The dev board consists of the ESP32-WROVER/ESP32-WROVER-B module, a Micro SD card, expansion interfaces, touch buttons and several function keys. It facilitates the quick and easy development of dual-mode (Bluetooth + Wi-Fi) audio solutions , also supporting one-key Wi-Fi configuration, a wake-up button, voice wake-up, voice recognition, cloud platform access, and an audio player.

The ESP32-LyraT smart audio board has the following features:

- Various mainstream, both lossy and lossless, compressed audio formats, including M4A, AAC, FLAC, OGG, OPUS, MP3, etc.
- One-key configuration and wake-up from the standby mode.
- SoftAP and Station mode.
- Various wireless protocols Wi-Fi 802.11b/g/n, Classic BT and BLE.
- A series of audio inputs, including Wi-Fi, BT-audio, DLNA, Line-in, etc.
- BLE network configuration, and smart network configuration with apps, such as WeChat.
- Two microphones for the development of near-field and far-field voice recognition applications.
- Peripherals for differentiated demands.

Figure 24 and 25 show the top view and bottom view of ESP32-LyraT.

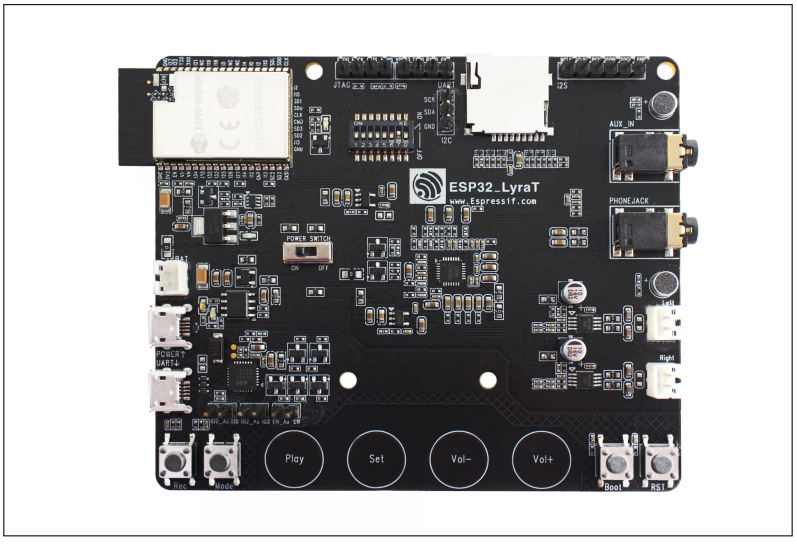

Figure 24: Top View of ESP32-LyraT

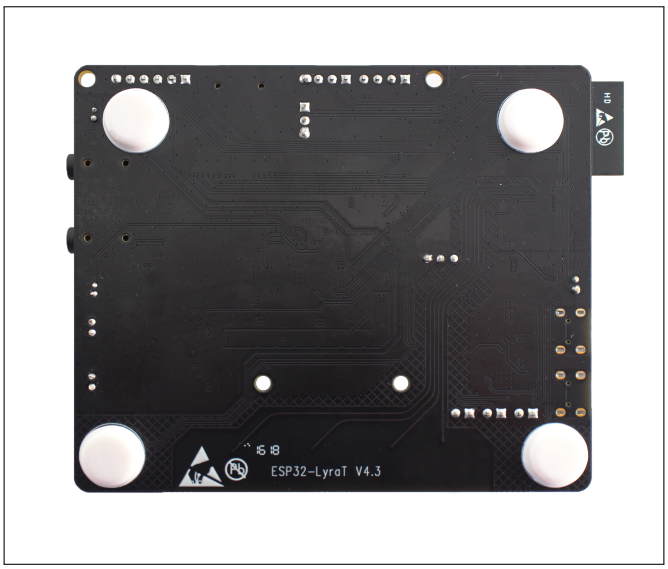

Figure 25: Bottom View of ESP32-LyraT

4.1.2 ESP32-LyraTD-MSC Audio Development Board

ESP32-LyraTD-MSC is designed for smart speakers and AI applications. This audio development board consists of two parts: the upper board, which provides a three-microphone array, function keys and LED lights; and the lower board, which integrates ESP32-WROVER-B, a MicroSemi Digital Signal Processing (DSP) chip, and a power management module. ESP32-LyraTD-MSC facilitates the quick and easy development of dual-mode (Bluetooth + Wi-Fi) audio solutions, as it supports one-key Wi-Fi network configuration, Acoustic Echo Cancellation (AEC), near/far-field voice wake-up, cloud platform access, voice recognition, wake-up interrupt and audio decoding.

The ESP32-LyraTD-MSC smart audio board has the following features:

- A lightweight, low-power and cost-effective smart audio solution.
- Access to multiple cloud platforms including DuerOS, Amazon, Tmall Genie, Turing, JD and iFLYTEK.
- HTTP live streaming, such as Internet radio and Ximalaya.
- Voice wake-up optimized with speech recognition and echo cancellation.
- Three digital MICs support far-field voice pick-up (from a distance of one to five meters).
- The dual-board design integrates an LED light strip and fully-functional buttons.
- Audio inputs over Wi-Fi, BT, DLNA, SD-Card.
- A variety of network configurations, such as Smartconfig, BLE, and Air-kiss.
- Wi-Fi 802.11b/g/n, Classic BT and BLE in the 2.4GHz band.
- Multiple audio formats including M4A, AAC, FLAC, OGG, OPUS, MP3, AMR.

ESP32-LyraTD-MSC's layout is shown in Figure 26.

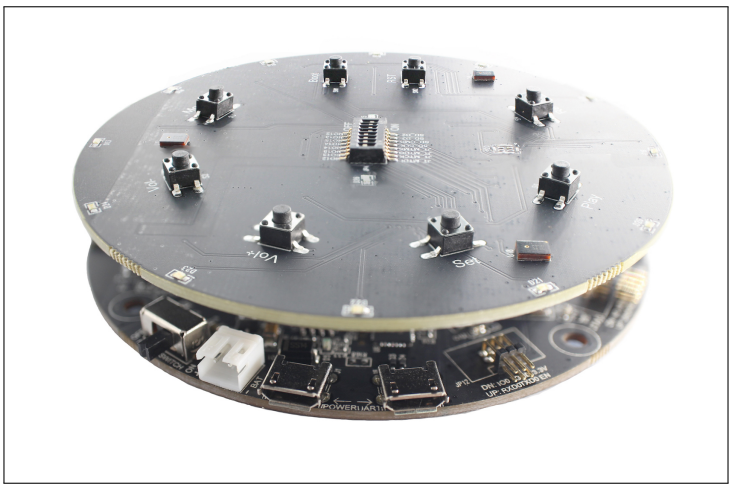

Figure 26: ESP32-LyraTD-MSC

Note:
Espressif provides design guidelines for audio products based on ESP32. For details please refer to *ESP32 Audio Design Guidelines*.

4.2 ESP32 Touch Sensor Application—ESP32-Sense Kit

The ESP32 touch sensor development kit, ESP32-Sense Kit, is used for evaluating and developing ESP32 touch sensor system. ESP32-Sense Kit consists of one motherboard and multiple daughterboards. The motherboard contains a display unit, a main control unit and a debug unit. The daughterboards have touch electrodes in different combinations or shapes, such as linear slider, wheel slider, matrix buttons and spring buttons, depending on the application scenarios. Users can design and add their own daughterboards for special usage cases.

The following image shows the whole ESP32-Sense development kit.

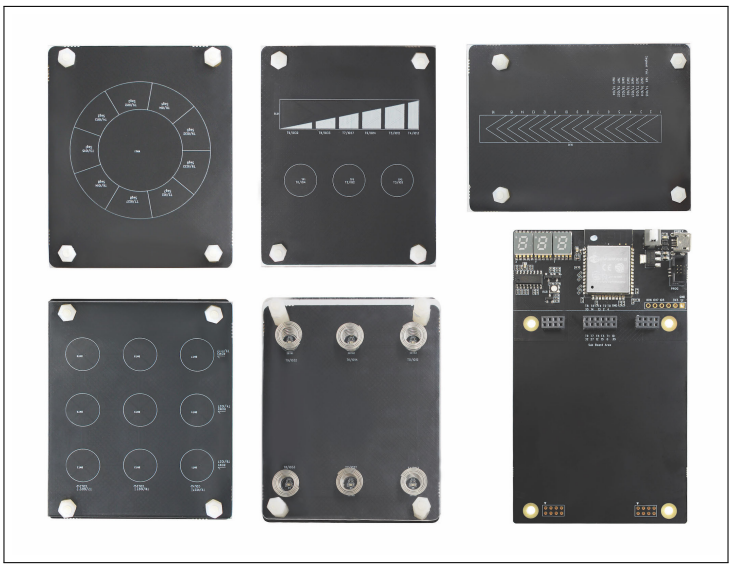

Figure 27: ESP32-Sense Kit

4.3 ESP-Mesh Application—ESP32-MeshKit

ESP32-MeshKit is an all-in-one smart-light development kit, which is developed based on ESP-Mesh technology. The development kit contains a number of ESP32-MeshKit-Lights and an ESP32-MeshKit-Sense board. ESP-Mesh is developed upon ESP-MDF, Espressif Mesh Development Framework.

ESP32-MeshKit-Lights are smart lights besed on ESP-Mesh. Users can control the lights either with the ESP-Mesh app, or the ESP32-MeshKit-Sense board that automatically switches on/off the lights by sensing the surrounding temperature. ESP32-MeshKit-Lights also supports secondary development.

Figure 28 shows an ESP32-MeshKit-Light.

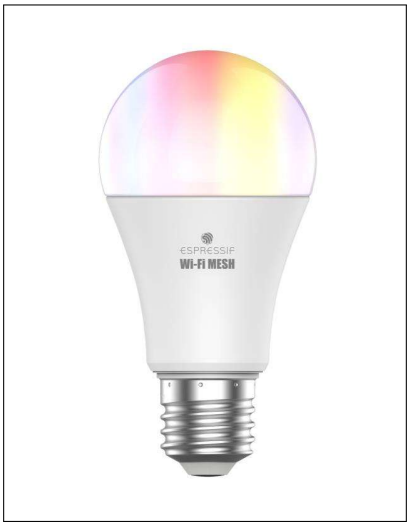

Figure 28: ESP32-MeshKit-Light

ESP32-MeshKit-Sense is a development board with an ESP32 module at its core. It integrates a temperature and humidity sensor and an ambient light sensor. The board can be connected to display screens. The integrated sensors on the board can automatically switch on/off the ESP32-MeshKit-Lights by sensing the surrounding environment. Apart from smart lights, the development board can also form a mesh network with other devices. In addition, the ESP32-MeshKit-Sense development board is a low-power sensing solution that can be used to detect the current consumption of ESP32 modules in a normal operation state or in sleep mode, when connected to different peripherals.

Figure 29 shows an ESP32-MeshKit-Sense development board.

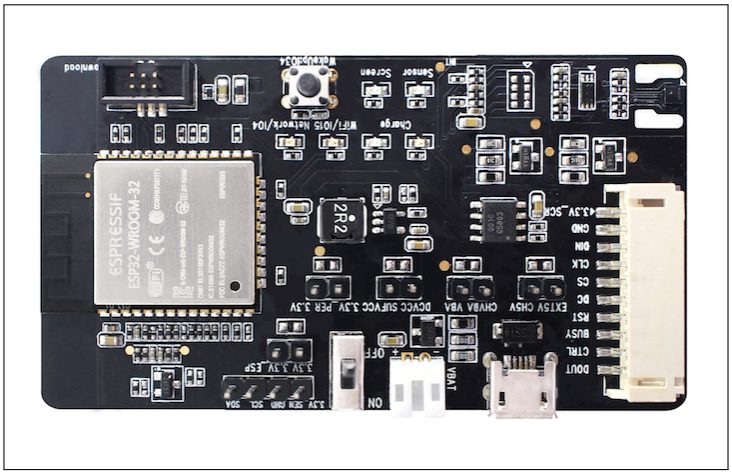

Figure 29: ESP32-MeshKit-Sense Development Board

Revision History

Date	Version	Release notes
2019.02	V2.6	Added two-layer PCB design guidelines in Sections Power Supply and RF; Updated Section UART; Updated the modules supported on dev boards ESP32-LyraT and ESP32-LyraTD-MSC according to the latest *Espressif Product Ordering Information*.
2018.11	V2.5	Updated Section Digital Power Supply: 2 kΩ resistor + 1 μF capacitor close to VDD_SDIO when it outputs 1.8 V is changed to 2 kΩ resistor + 4.7 μF capacitor.
2018.09	V2.4	Updated Section 2.1.4.1: External Clock Source; Updated Section 2.1.4.2: RTC.
2018.08	V2.3	Updated Section 2.1.7: External Capacitor.
2018.07	V2.2	Changed the official name of ESP-WROOM-32 into ESP32-WROOM-32; Changed the official name of ESP-WROOM-32D into ESP32-WROOM-32D. Updated Chapter 2: Schematic Checklist and PCB Layout Design; Added new modules and applications based on ESP32.
2018.03	V2.1	Deleted sections introducing protocols, applications, block diagram and pin description of ESP32, for information of which please refer to *ESP32 Datasheet*; Updated all figures and description of schematics and PCB layout in Chapter 2 ; Added Section 2.1.6 ADC and 2.1.8 UART; Updated Section 2.2, and added description about Positioning a ESP32 Module on a Base Board in it. Updated the value of current ripple in Section 2.2.3.1. Updated Section ESP32-DevKitC Development Board. Added Section ESP32-PICO-D4 Module. Added Section ESP32-WROOM-32D Module. Added Section ESP32-WROOM-32U Module. Added Section ESP32-PICO-KIT Mini Development Kit. Deleted orginal Section 2.1.6 ESP32-PICO-KIT Mini Development Kit.
2017.08	V2.0	Changed the transmitting power to +12 dBm; the sensitivity of NZIF receiver to -97 dBm in Section Bluetooth; Added a note to Table Pin Description; Added Section 2.2.1.8 Touch Sensor; Updated Chapter 3 Hardware Development; Updated Section 4.1.1.
2017.06	V1.9	Changed the input power supply range of CPU/RTC IO to 1.8V ~ 3.6V; Updated Section 2.1.1.1 Digital Power Supply.
2017.06	V1.8	Updated Section 2.1.2.1 Power-on Sequence; Updated Section 2.1.4.1 External Clock Source (Compulsory); Added a link to ESP32 Pin Lists; Added Documentation Change Notification.
2017.05	V1.7	Added a note to Section ESP32-WROOM-32 Overview.
2017.05	V1.6	Updated Figure ESP32-WROOM-32 Pin Layout; Added a note in Section Strapping Pins.

Date	Version	Release notes
2017.04	V1.5	Added the ESP-WROOOM-32 module's dimensional tolerance.
2017.04	V1.4	Updated Section Strapping Pins; Updated Figure ESP32 Pin Layout (for QFN 5*5); Updated Figure ESP32-WROOM-32 Module; Updated Figure ESP32-DevKitC Pin Layout.
2017.03	V1.3	Updated the notice to Table ESP32 Pin Description; Added a note to Table ESP32-WROOM-32 Pin Definitions.
2017.03	V1.2	Updated Chapter Overview; Updated Figure Function Block Diagram; Updated Chapter Pin Definitions; Updated Section Power Supply; Updated Section RF; Updated Figure ESP32-WROOM-32 Pin Layout; Updated Table ESP32-WROOM-32 Pin Definitions; Updated Section Notes.
2016.12	V1.1	Updated Table UART to Wi-Fi Smart Device.
2016.12	V1.0	First release.