ESP32 Hardware Design Guidelines

Espressif Systems

January 12, 2018

About This Guide

The guidelines outline recommended design practices when developing standalone or add-on systems based on the ESP32 series of products, including ESP32, the ESP-WROOM-32 module, and ESP32-DevKitC — the development board.

Related Resources

For additional documentation and resources on ESP32, please visit Espressif website: ESP32 Resources.

Release Notes

Date	Version	Release notes
2016.12	V1.0	First release.
2016.12	V1.1	Updated Table 2.
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 ESP-WROOM-32 Pin Layout; Updated Table ESP-WROOM-32 Pin Definitions; Updated Section Notes.
2017.03	V1.3	Updated the notice to Table ESP32 Pin Description; Added a note to Table ESP-WROOM-32 Pin Definitions.
2017.04	V1.4	Updated Section Strapping Pins; Updated Figure ESP32 Pin Layout (for QFN 5*5); Updated Figure ESP-WROOM-32 Module; Updated Figure ESP32-DevKitC Pin Layout.
2017.04	V1.5	Added the ESP-WROOOM-32 module's dimensional tolerance.
2017.05	V1.6	Updated Figure ESP-WROOM-32 Pin Layout; Added a note in Section Strapping Pins.
2017.05	V1.7	Added a note to Section ESP-WROOM-32 Overview.
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.
2017.06	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.
2017.08	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 4.2.

Date	Version	Release notes
2018.01	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 2.1.7 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.

Documentation Change Notification

Espressif provides email notifications to keep customers updated on changes to technical documentation. Please subscribe here.

Certification

Download certificates for Espressif products from here.

Disclaimer and Copyright Notice

Contents

List of Tables

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.

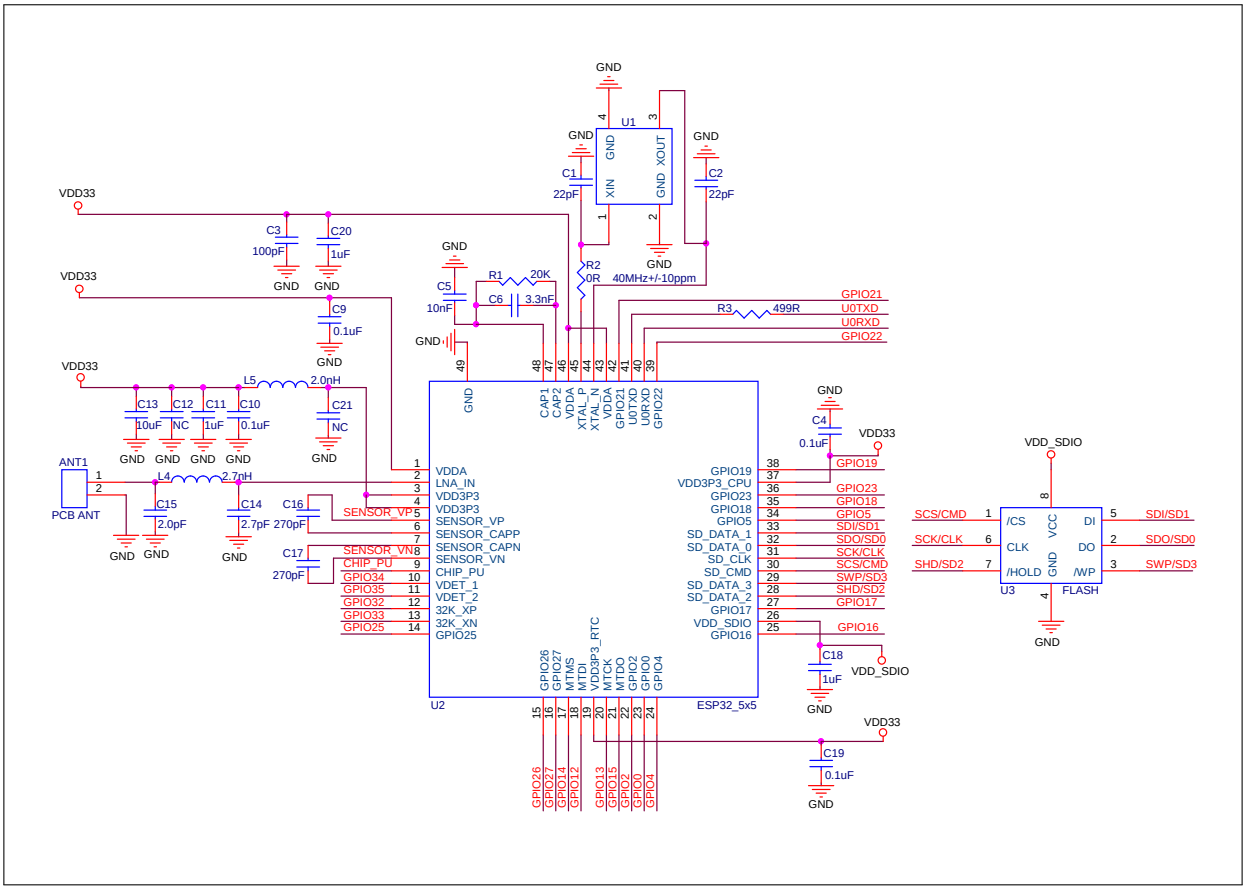

Figure 1: ESP32 Schematics

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

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

- Crystal oscillator
- RF
- External capacitors
- UART

2.1.1 Power Supply

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 1.8V ~ 3.6V. We recommend adding extra filter capacitors of 0.1 μF close to the digital power supply pins.

The voltage of VDD_SDIO is produced by the internal LDO. It can be used as the power supply for the external circuitry, with a maximum current of about 40 mA when using the 3.3V LDO. When the VDD_SDIO outputs 1.8V, the value of GPIO12 should be set to 1 when the chip boots. The user can add a 1 μF filter capacitor close to VDD_SDIO. When the VDD_SDIO outputs 3.3V, the value of GPIO12 is 0 (default) when the chip boots and it is recommended that users add 2 kΩ resistor to ground and a 1 μF capacitor close to VDD3P3_RTC.

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

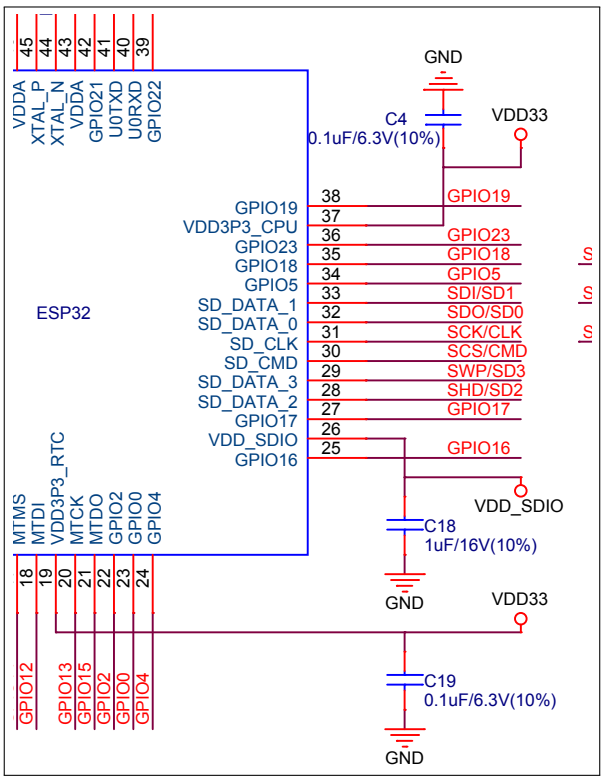

Figure 2: ESP32 Digital Power Supply Pins

2.1.1.2 Analog Power Supply

Pin1, Pin43 and Pin46 are the analog power supply pins. Pin3 and Pin4 are the power supply pins for the power amplifiers. 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 to near the power pin so as to suppress high-frequency harmonics. The inductor's rated current is preferably 500 mA and above.

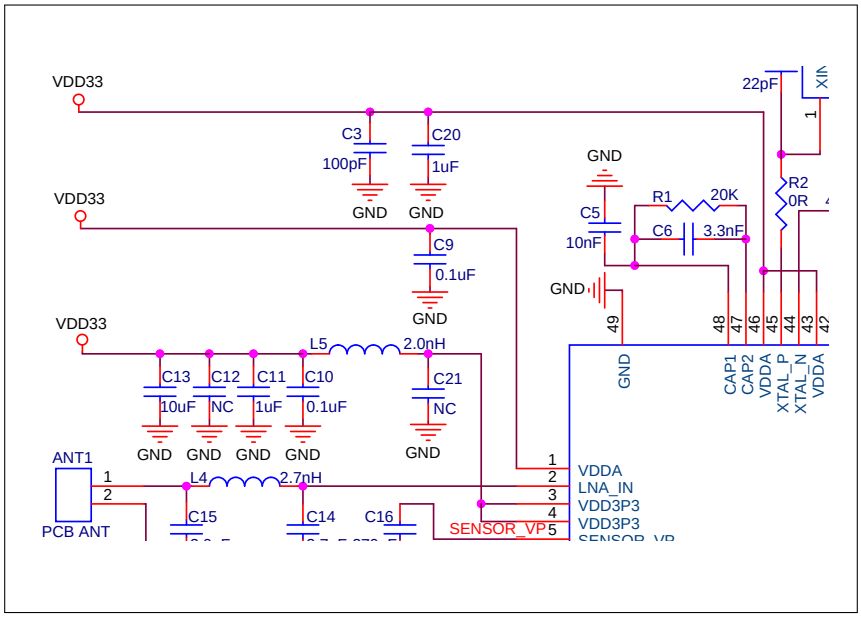

Figure 3: ESP32 Analog Power Supply Pins

Notice:

- When using a single power supply, the operating voltage for ESP32 ranges from 2.3V to 3.6V. The recommended voltage of the power supply for ESP32 is 3.3V, 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 3.3V 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) by time T after the 3.3V rails have been brought up. The recommended delay time (T) is given by the parameter of the RC(R = 10 kΩ, C = 0.1 μF) circuit. For reference design, please refer to Figure **ESP-WROOM-32 Peripheral Schematics** in the *ESP-WROOM-32 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 (R = 10 kΩ, C = 0.1 μF) circuit is required.

2.1.2.2 Reset

CHIP_PU serves as the reset pin of ESP32. ESP32 will reset when CHIP_PU is held low and the input level is below 0.6V and stays for at least 200 μs. 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 you select the appropriate flash according to the power voltage on VDD_SDIO. Users can add a serial resistor to Pin21 SD_CLK and connect it to the flash CLK pin.

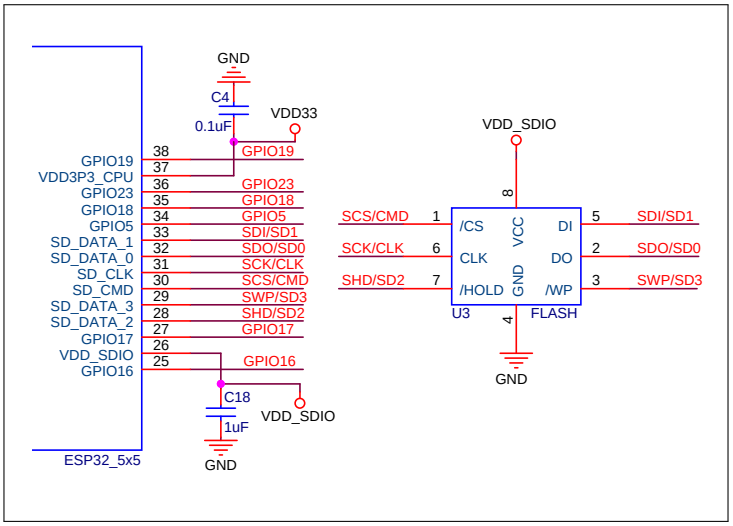

Figure 4: ESP32 Flash

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)

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 0Ω on the XTAL_P clock trace. Note that the accuracy of the selected crystal is ± 10 PPM.

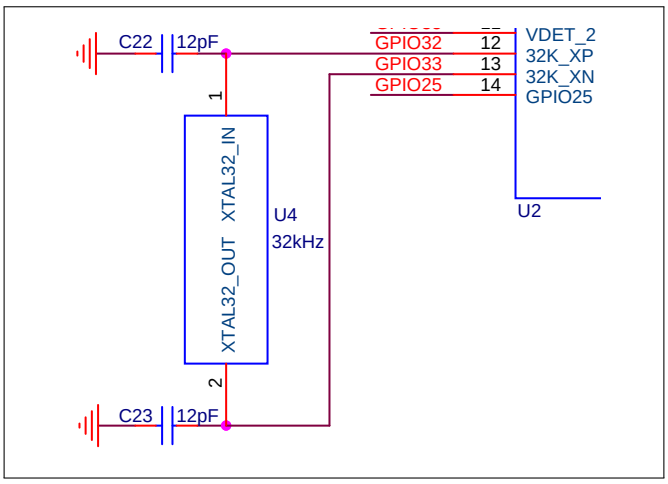

Figure 5: ESP32 Crystal Oscillator

Notice:

Defects in the craftsmanship of the crystal oscillators (for example, high frequency deviation) 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 oscillator to act as the RTC sleep clock.

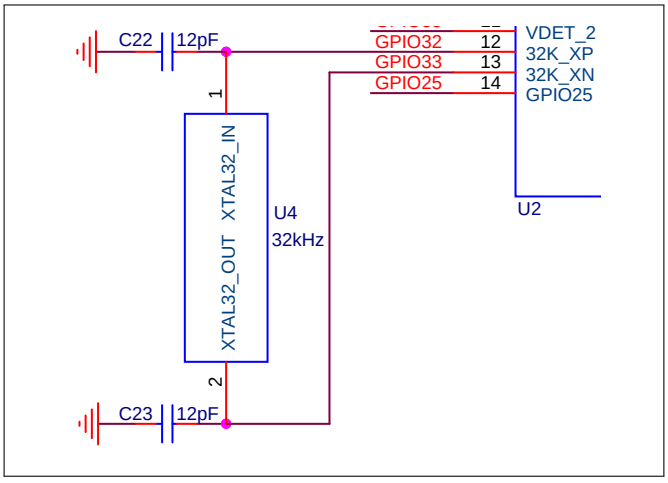

Figure 6: ESP32 Crystal Oscillator (RTC)

Notice:

If the RTC source is not required, then Pin12 32K_XP and Pin13 32K_XN can be used as GPIOs.

2.1.5 RF

The output impedance of the RF pins of ESP32 (QFN 6*6) and ESP32 (QFN 5*5) are (30+j10) 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.

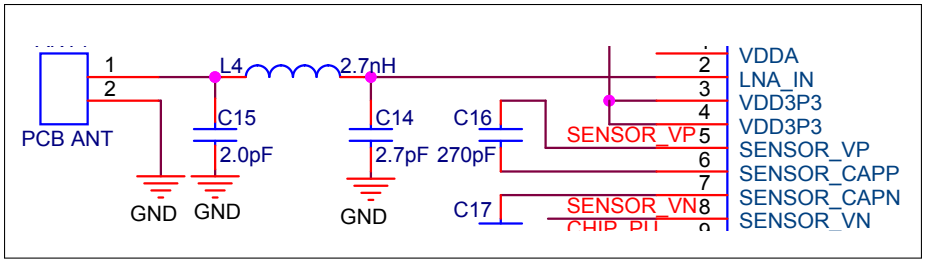

Figure 7: ESP32 RF Matching Schematics

Note:

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

2.1.6 External Capacitor

The schematics of Pin47 CAP2 and Pin48 CAP1 is shown in Figure 8. C5 (10 nF) that connects to CAP1 should be of high precision. For the RC circuit between CAP1 and CAP2 pins, please refer to Figure 8. Removing the RC circuit may slightly affect ESP32 in Deep-sleep mode.

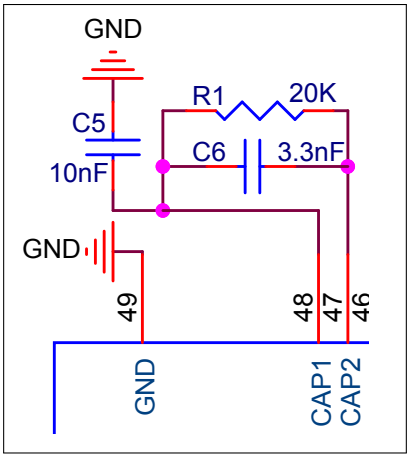

Figure 8: ESP32 External Capacitor

2.1.7 UART

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

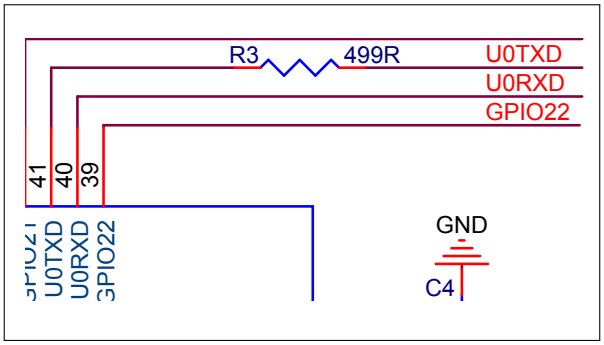

Figure 9: ESP32 UART

2.2 PCB Layout Design

This chapter introduces the key points of designing ESP32 PCB layout with the example of ESP-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.

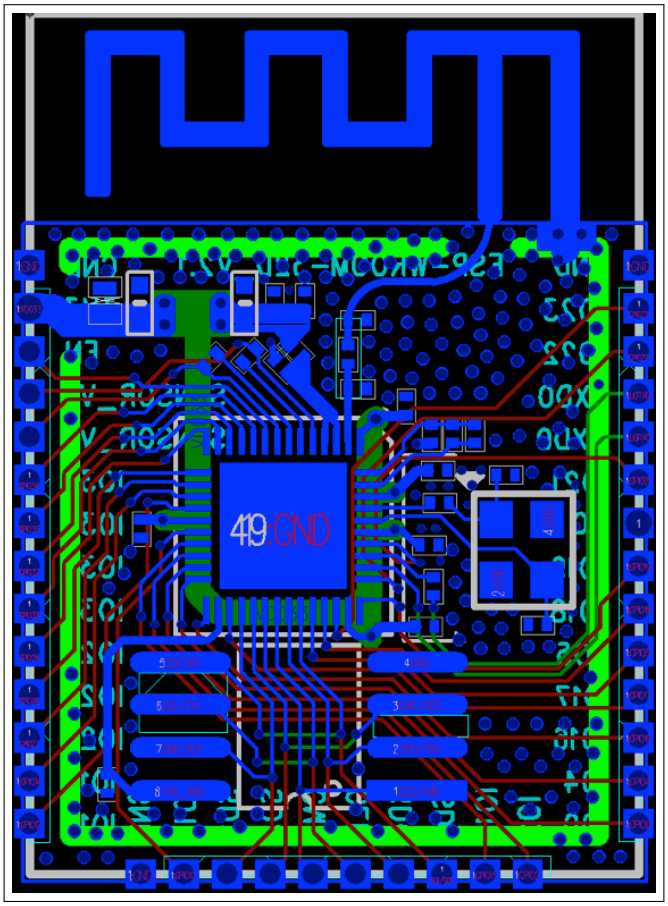

Figure 10: ESP32 PCB Layout

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 four-layer PCB design.

- The first layer is the TOP layer for signal traces and components.

- The second layer is the BOTTOM layer, where power traces are routed. Placing any components on this layer is not recommended. Do not route any power or signal traces under or around the RF and crystal oscillator, so that there is a complete GND plane, which is connected to the Gound Pad at the bottom of the chip.

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.

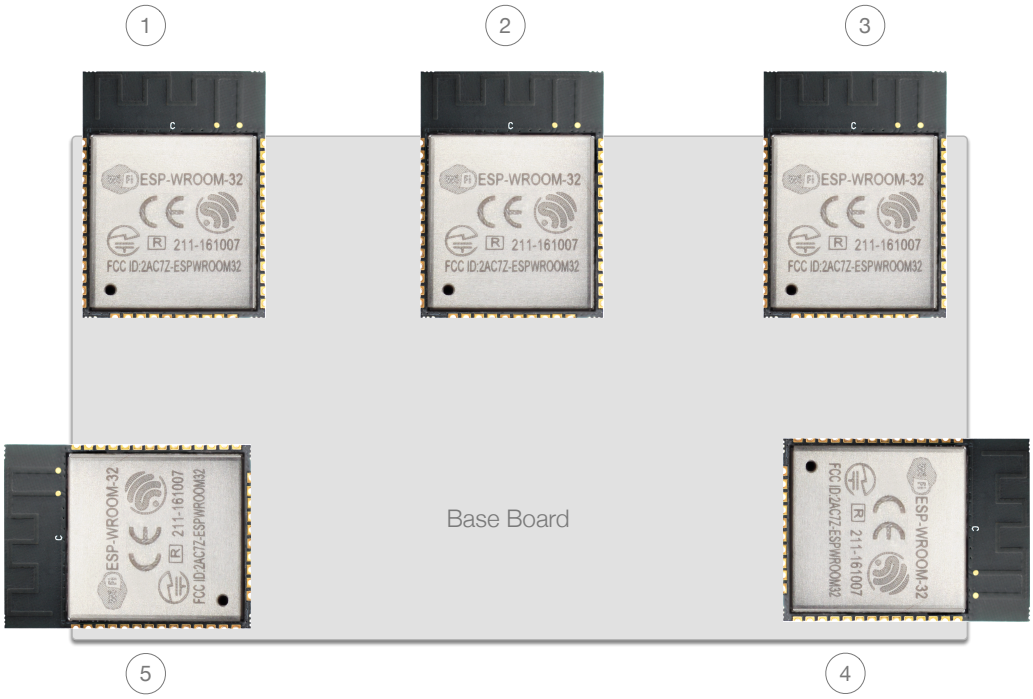

Figure 11: ESP32 Module Antenna Position on Base Board

Note:

As is shown in Figure 11, 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 the Figure 12:

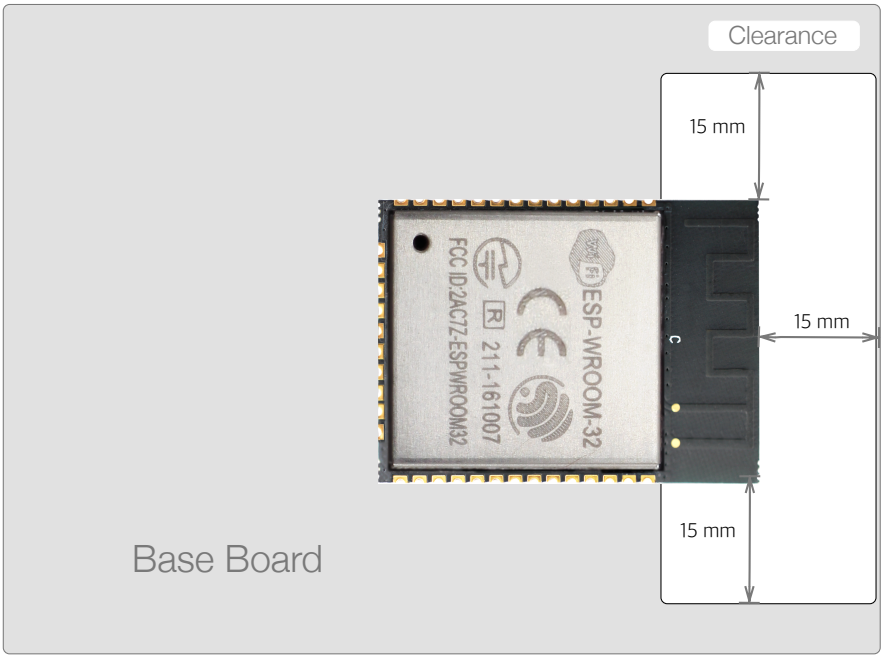

Figure 12: Keepout Zone for ESP32 Module's Antenna on the Base Board

2.2.1.3 Power Supply

The 3.3V power traces are highlighted in yellow in Figure 13. The width of these power traces should be larger than 20 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 13 shows, C13 (10 μF capacitor) is placed by the 3.3V 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.

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 center ground pad at the bottom of the chip should be connected to ground plane through at least 9 ground vias.

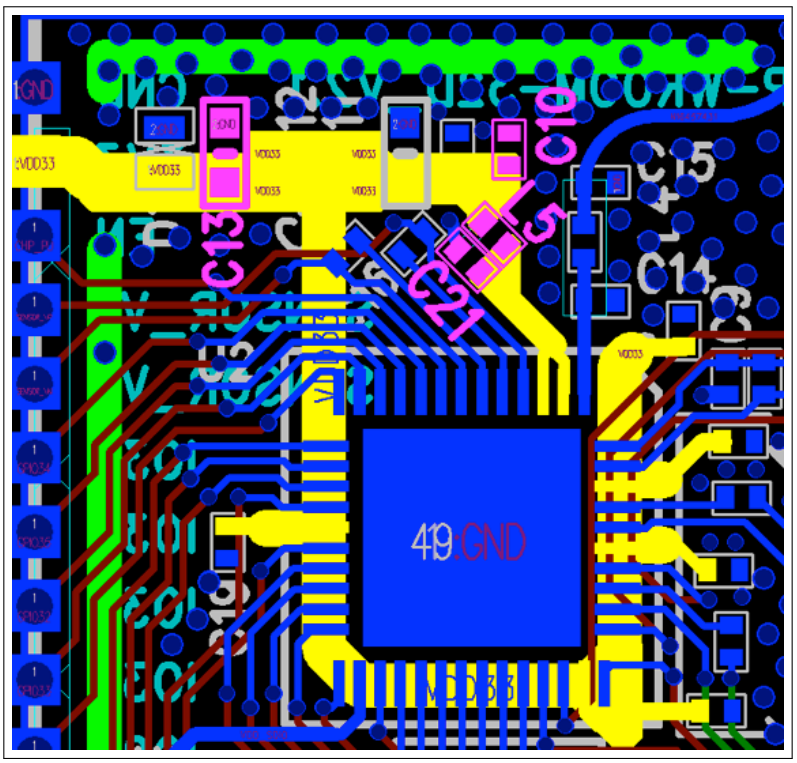

Figure 13: ESP32 Power Supply Design

2.2.1.4 Crystal Oscillator

For the design of the crystal oscillator section, please refer to Figure 14. 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 that 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.

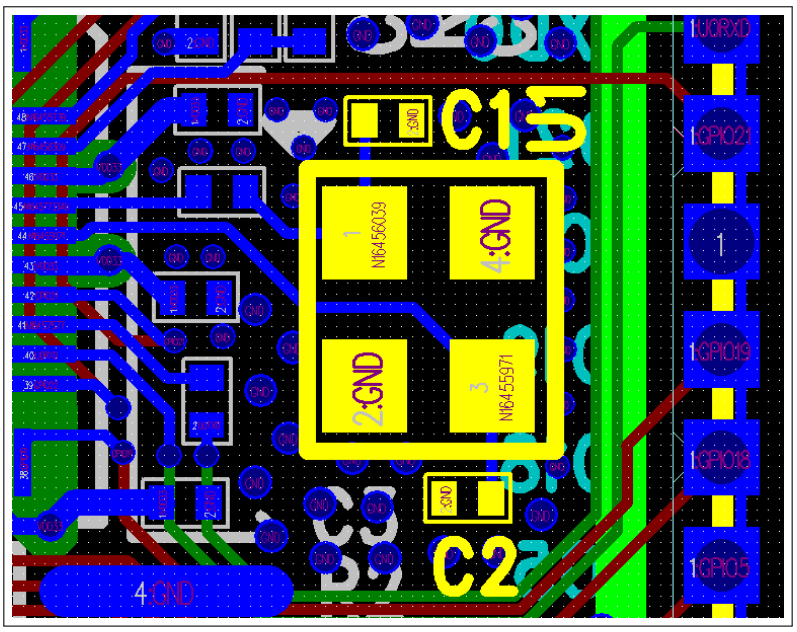

Figure 14: ESP32 Crystal Oscillator Layout

2.2.1.5 RF

The characteristic RF impedance must be 50Ω. The ground plane on the adjacent layer needs to be complete. Make sure you 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-A antenna.

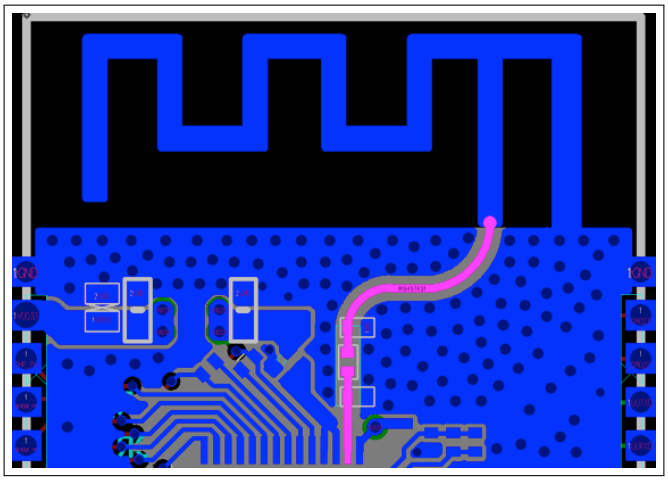

Figure 15: ESP32 RF Layout

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 as possible to the chip so that the U0TXD traces on the top layer are as short as possible.

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.

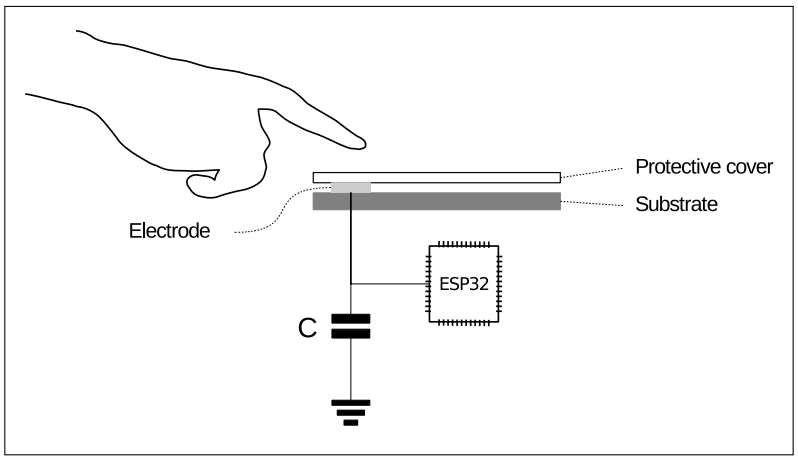

Figure 16: A Typical Touch Sensor Application

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.

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.

Note:
The examples illustrated in Figure 17 are not of actual scale. It is suggested that users use a human fingertip as reference.

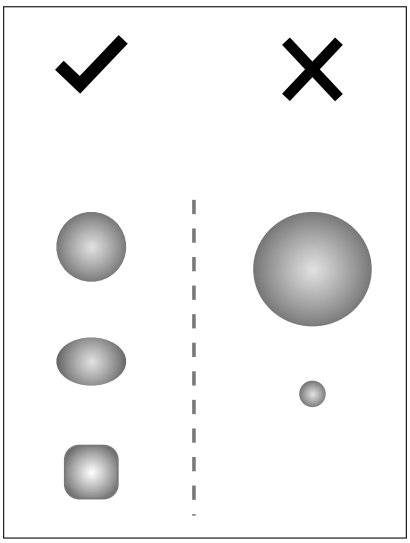

Figure 17: Electrode Pattern Requirements

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 sensor-to-ground gap (S) should not be less than 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.

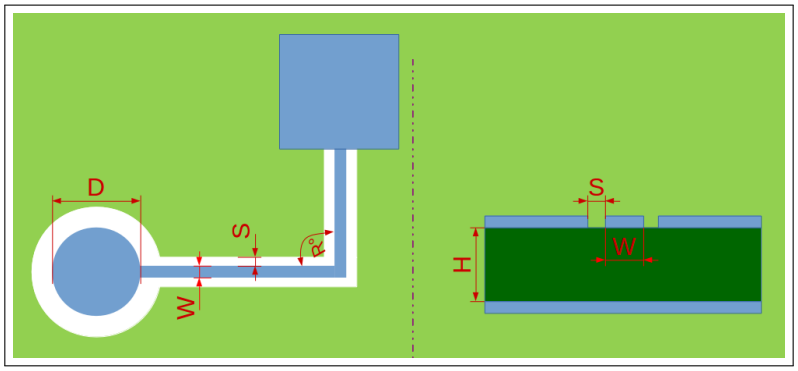

Figure 18: Sensor Track Routing Requirements

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

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.

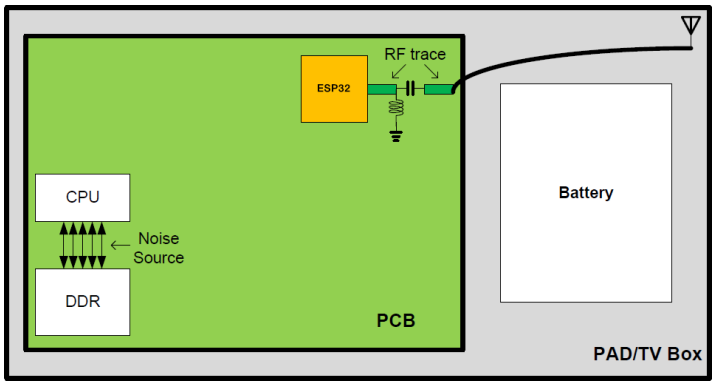

Figure 19: PAD/TV Box Layout

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 19, 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 Chapter 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 Chapter 2.2 for details.

3. Hardware Development

Note:
For more information on ESP32 modules, please refer to the webpage Espressif Modules.

3.1 ESP-WROOM-32 Module

Espressif provides users with an SMD module, the ESP-WROOM-32. This module has been adjusted to achieving the optimum RF performance.

The size of the module is 18±0.2 x 25.5±0.2 x 3.1±0.15 (mm). The flash used is in an SOP8-208 mil package. The on-board PCB antenna has a gain of 2 dBi.

Figure 20 shows the front and the rear side of the module.

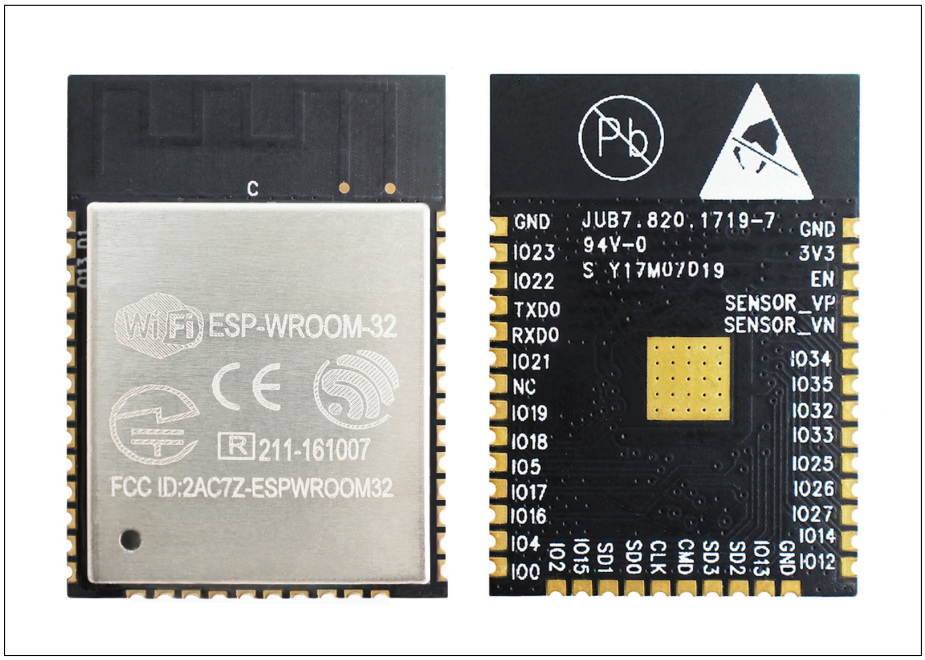

Figure 20: ESP-WROOM-32 Module

For more information on the module's pin definition, physical dimensions, schematics, etc, please refer to ESP-WROOM-32 Datasheet.

3.2 ESP32-WROVER Module

ESP32-WROVER is another ESP32-based module. Compared to ESP-WROOM-32, ESP32-WROVER has an additional SPI Pseudo-static RAM (PSRAM) of 32 Mbits. As such, ESP32-WROVER features both 4 MB external SPI flash and 4 MB external PSRAM.

The ESP32-WROVER module has a PCB antenna, while the ESP32-WROVER-I uses an IPEX antenna.

Figure 21 shows the front and the rear side of the module.

For more information on the module's pin definition, physical dimensions, schematics, etc, please refer to ESP32-WROVER Datasheet.

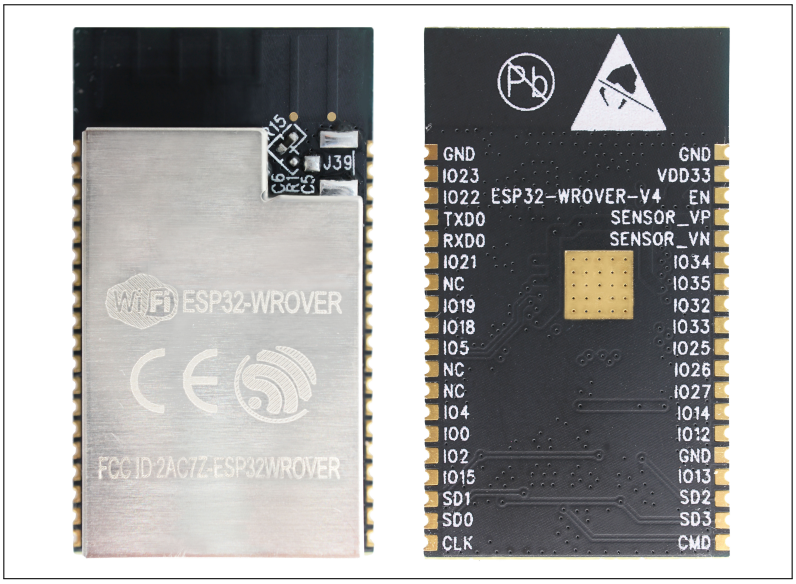

Figure 21: ESP32-WROVER Module

3.3 Notes on Using Modules

- The module uses one single pin as the power supply pin. Users can connect the module to a 3.3V power supply. The 3.3V 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.
- The SMD Module features two working modes: the UART Download mode and the Flash Boot mode. In the UART Download mode, firmware can be downloaded into the flash memory or the internal memory by configuring the flash download tool. If the firmware is burnt into the internal memory, it can only run once and when the module is powered on. When the module is powered down, the internal memory will clear up. However, if the firmware is burnt and stored into the flash, it will be recalled at any time.
- 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.

3.4 ESP32-DevKitC Development Board

ESP32-DevKitC is a low-footprint, minimal system development board which is powered by ESP-WROOM-32. The dimensions of the board are shown in Figure 22.

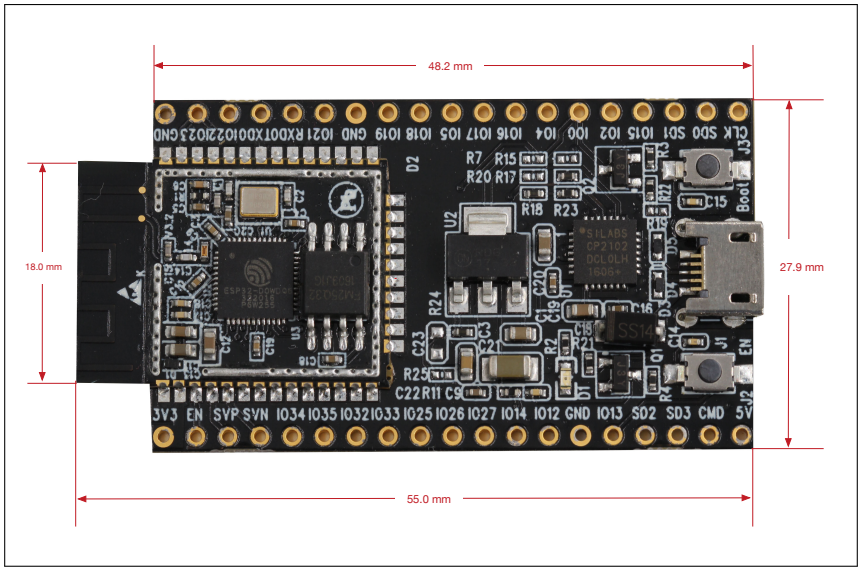

Figure 22: Front Side of ESP32-DevKitC

All pins of ESP-WROOM-32 are led out to the pin headers on both sides for easy interfacing. ESP32-DevKitC features all the functions that are supported by ESP32. Users can connect these pins to peripherals as needed.

For more information on this board's layout, schematics, etc, please refer to ESP32-DevKitC Getting Started Guide.

3.5 ESP-WROVER-KIT Development Board

The ESP-WROVER-KIT is another development board built around ESP32. This board is compatible with ESP32 modules, including the ESP-WROOM-32 and ESP32-WROVER. The ESP-WROVER-KIT features support for an LCD and MicroSD card. The I/O pins have been broken out from the ESP32 module for easy extension. The board carries an advanced multi-protocol USB bridge (the FTDI FT2232HL), enabling developers to use JTAG directly to debug ESP32 through the USB interface. The development board makes secondary development easy and cost-effective.

The board's layout is shown in Figure 23 and Figure 24.

For more information on this board's layout, schematics, etc., please refer to ESP-WROVER-KIT Getting Started Guide.

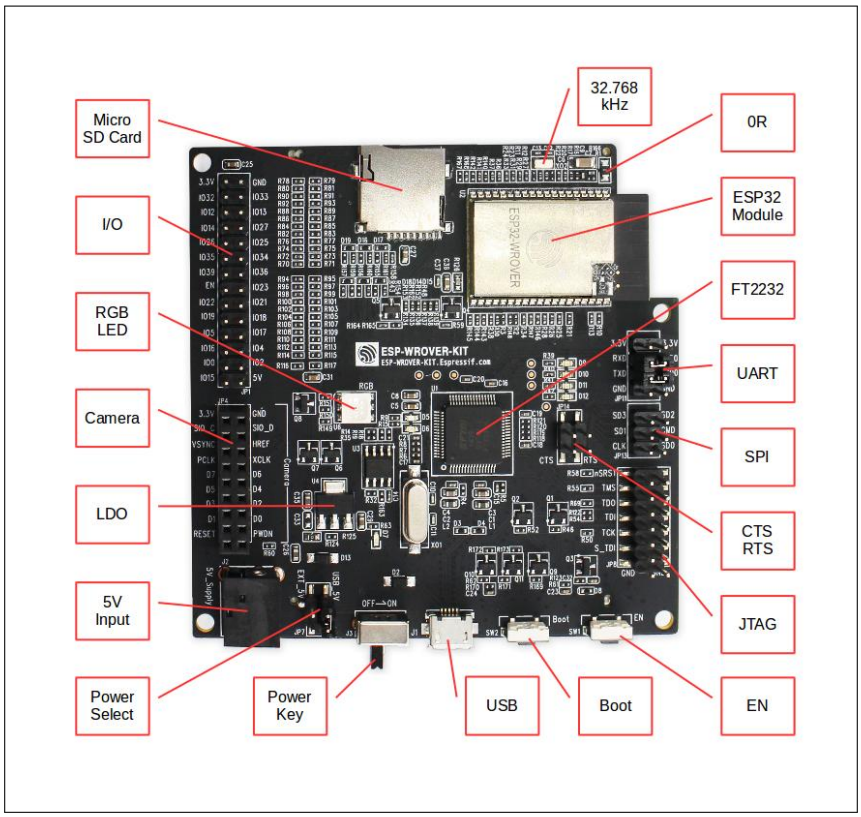

Figure 23: Front Side of the ESP-WROVER-KIT Layout

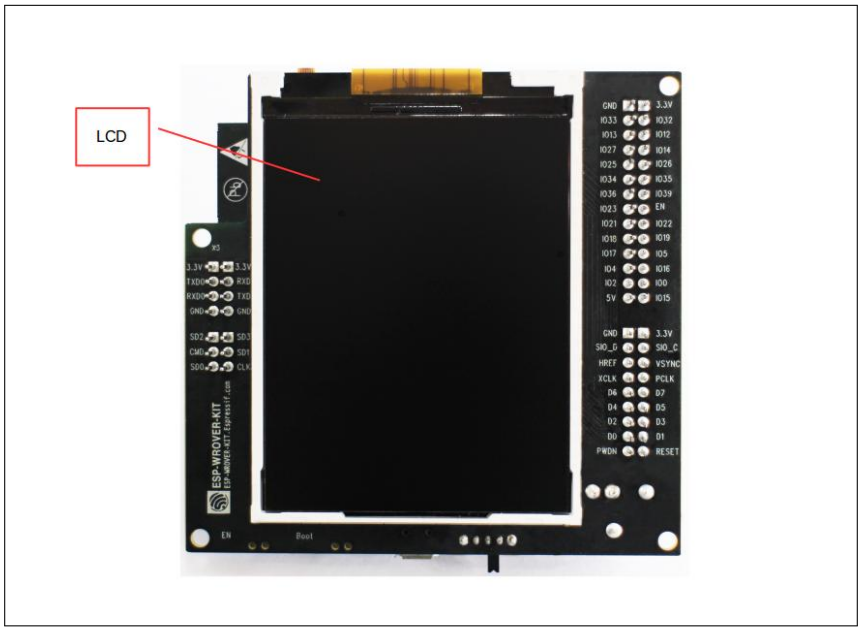

Figure 24: Rear Side of the ESP-WROVER-KIT Layout

4. Applications

4.1 UART to Wi-Fi Smart Device

The two UART interfaces are defined in Table 2.

Table 2: Pin Definition of UART Interfaces

Categories	Pin Definition	Function
UART0	(Pin34) U0RXD + (Pin35) U0TXD	Used for printing logs.
UART1	(Pin25) U1RXD + (Pin27) U1TXD	Used for receiving and sending commands.

Application example: ESP32-DevKitC (please see Section 3.4 ESP32-DevKitC).

4.2 ESP32-LyraT Smart Audio Platform

ESP32-LyraT is a smart audio platform designed with voice-recognition technology, targeting the IoT market. ESP32-LyraT is based on the highly-integrated ESP32-WROVER module which features not only a 4 MB SPI flash, but also a 4 MB PSRAM. With its ESP32 dual-core processors, Wi-Fi+BT capabilities and high integration, ESP32-LyraT delivers a rapid product-development platform for systems of artificial intelligence, voice and image recognition, wireless audio and smart-home networks.

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

- Support for multiple audio input sources: Wi-Fi, BT audio, AirPlay, DLNA, line-in, etc.
- Support for dual microphone pickup, near-field and far-field voice recognition.
- Support for mainstream lossless audio formats: ALAC, AAC, FLAC, OPUS, MP3, WAV, OGG, etc.
- Support for complete wireless protocols: Wi-Fi 802.11b/g/n, Classic BT, BLE, etc.
- Support for multiple networking protocols: BLE, WeChat, etc.
- Support for various interfaces with high extensibility: network interface, touch buttons, TFT screen, Camera interface, etc.
- Support for multiple cloud platforms: DuerOS, Ximalaya FM, DeepBrain, etc.