

ESP32 Hardware Design Guidelines



Espressif Systems

March 9, 2017

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 the website: [ESP32 Resources](#).

Release Notes

Date	Version	Release notes
2016.12	V1.0	First release.
2016.12	V1.1	Updated Table 4.
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 3; Updated Section Notes.

Disclaimer and Copyright Notice

Information in this document, including URL references, is subject to change without notice. THIS DOCUMENT IS PROVIDED AS IS WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR 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.

Copyright © 2017 Espressif Inc. All rights reserved.

Contents

1 Overview

1.1 Basic Protocols	6
1.1.1 Wi-Fi	6
1.1.2 Bluetooth	6
1.2 Application	7
1.3 Function Block Diagram	8

2 Pin Definitions

2.1 Pin Layout	9
2.2 Pin Description	10
2.3 Strapping Pins	12

3 Schematics and PCB Layout Design

3.1 Schematics	14
3.1.1 Power Supply	15
3.1.1.1 Digital Power Supply	15
3.1.1.2 Analog Power Supply	16
3.1.2 Power-on Sequence and System Reset	16
3.1.2.1 Power-on Sequence	16
3.1.2.2 Reset	16
3.1.3 Flash	17
3.1.4 Crystal Oscillator	17
3.1.4.1 External Clock Source (compulsory)	17
3.1.4.2 RTC (Optional)	18
3.1.5 RF	18
3.1.6 Sensor External Sampling Capacitor	19
3.1.7 External Capacitor	19
3.2 PCB Layout	19
3.2.1 Standalone ESP32 Module	20
3.2.1.1 General Principles of PCB Layout	20
3.2.1.2 Power Supply	20
3.2.1.3 Crystal Oscillator	20
3.2.1.4 RF	21
3.2.2 ESP32 as a Slave Device	22
3.2.3 Typical Layout Problems and Solutions	22
3.2.3.1 Q: The current ripple is not large, but the Tx performance of RF is rather poor.	22
3.2.3.2 Q: The power ripple is small, but RF Tx performance is poor.	23
3.2.3.3 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.	23
3.2.3.4 Q: Tx performance is not bad, but the Rx sensitivity is low.	23

4 Hardware Development

4.1 ESP-WROOM-32 Module	25
4.1.1 Overview	25

4.1.2	Pin Definition	26
4.1.3	Notes	27
4.2	ESP32-DevKitC	29
4.2.1	Overview	29
4.2.2	Schematics	30
4.2.2.1	Power Schematics	30
4.2.2.2	USB-UART Schematics	30
5	Applications	31
5.1	UART to Wi-Fi Smart Device	31
5.2	ESP32-Lyra Smart Audio Platform	31

List of Tables

1	Pin Description	10
2	Strapping Pins	13
3	ESP-WROOM-32 Pin Definitions	26
4	Pin Definition of UART Interfaces	31

List of Figures

1	Function Block Diagram	8
2	ESP32 Pin Layout (for QFN 6*6)	9
3	ESP32 Pin Layout (for QFN 5*5)	10
4	ESP32 Schematics	14
5	ESP32 Digital Power Supply Pins	15
6	ESP32 Analog Power Supply Pins	16
7	ESP32 Flash	17
8	ESP32 Crystal Oscillator	18
9	ESP32 Crystal Oscillator (RTC)	18
10	ESP32 RF Matching Schematics	18
11	ESP32 Sensor External Sampling Capacitor	19
12	ESP32 External Capacitor	19
13	ESP32 PCB Layout	20
14	ESP32 Crystal Oscillator Layout	21
15	ESP32 RF Layout	21
16	PAD/TV Box Layout	22
17	ESP-WROOM-32 Module	25
18	ESP-WROOM-32 Pin Layout	26
19	Top view of ESP32-DevKitC	29
20	ESP32-DevKitC Pin Layout	29
21	Power Schematics of ESP32-DevKitC	30
22	USB-UART Schematics	30

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 performance and RF performance in a wide variety of applications and different power profiles, with robustness, versatility and reliability.

ESP32 is a Wi-Fi plus Bluetooth System-on-a-Chip (SoC). With only 16 external components, it has the optimal level of integration in the industry. It integrates the complete transmit/receive RF functionality including the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management, and advanced calibration circuitries that allow the solution to dynamically adjust itself to external circuit imperfections. As such, the mass production of ESP32-based solutions does not require expensive and specialized Wi-Fi testing 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](#).

1.1 Basic Protocols

1.1.1 Wi-Fi

- 802.11 b/g/n/e/i
- 802.11 n (2.4 GHz), up to 150 Mbps
- 802.11 e: QoS for wireless multimedia technology
- WMM-PS, UAPSD
- A-MPDU and A-MSDU aggregation
- Block ACK
- Fragmentation and defragmentation
- Automatic Beacon monitoring/scanning
- 802.11 i security features: pre-authentication and TSN
- Wi-Fi Protected Access (WPA)/WPA2/WPA2-Enterprise/Wi-Fi Protected Setup (WPS)
- Infrastructure BSS Station mode/SoftAP mode
- Wi-Fi Direct (P2P), P2P Discovery, P2P Group Owner mode and P2P Power Management
- UMA compliant and certified
- Antenna diversity and selection

1.1.2 Bluetooth

- Compliant with Bluetooth v4.2 BR/EDR and BLE specification
- Class-1, class-2 and class-3 transmitter without external power amplifier
- Enhanced power control
- +10 dBm transmitting power
- NZIF receiver with -98 dBm sensitivity

- Adaptive Frequency Hopping (AFH)
- Standard HCI based on SDIO/SPI/UART
- High-speed UART HCI, up to 4 Mbps
- BT 4.2 controller and host stack
- Service Discover Protocol (SDP)
- General Access Profile (GAP)
- Security Manage Protocol (SMP)
- Bluetooth Low Energy (BLE)
- ATT/GATT
- HID
- All GATT-based profile supported
- SPP-Like GATT-based profile
- BLE Beacon
- A2DP/AVRCP/SPP, HSP/HFP, RFCOMM
- CVSD and SBC for audio codec
- Bluetooth Piconet and Scatternet

1.2 Application

- Generic low-power IoT sensor hub
- Generic low-power IoT loggers
- Video streaming from camera
- Over The Top (OTT) devices
- Music players
 - Internet music players
 - Audio streaming devices
- Wi-Fi-enabled toys
 - Loggers
 - Proximity sensing toys
- Wi-Fi-enabled speech recognition devices
- Audio headsets
- Smart power plugs
- Home automation
- Mesh network
- Industrial wireless control

- Baby monitors
- Wearable electronics
- Wi-Fi location-aware devices
- Security ID tags
- Healthcare
 - Proximity and movement monitoring trigger devices
 - Temperature sensing loggers

1.3 Function Block Diagram

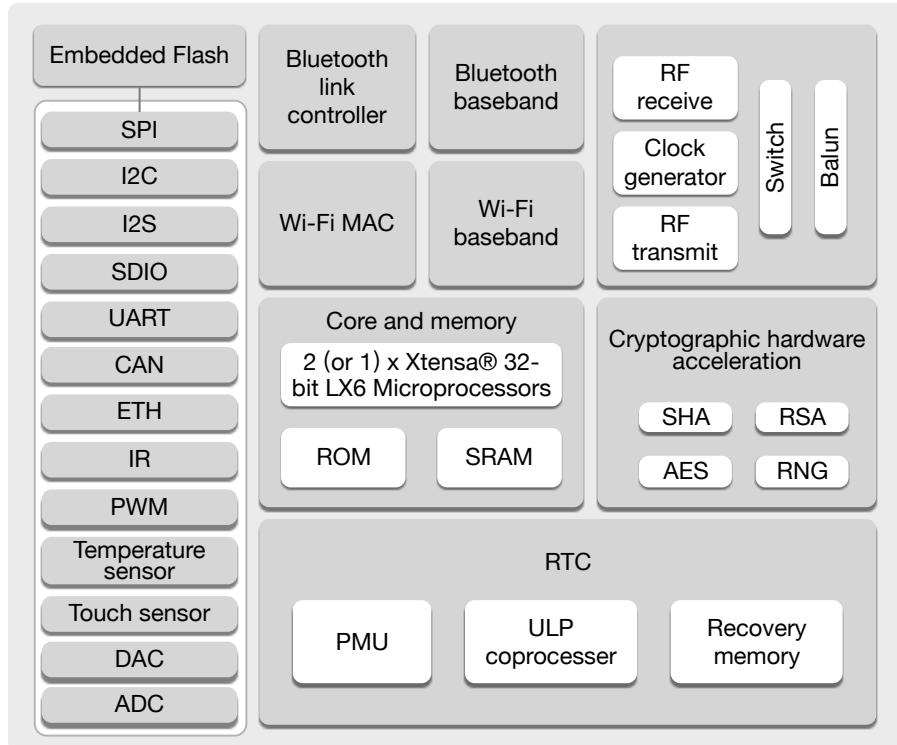


Figure 1: Function Block Diagram

Note:

Products in the ESP32 series differ from each other in terms of the number of CPUs they have and their support for embedded flash. For details, please refer to [ESP32 Datasheet](#).

2. Pin Definitions

2.1 Pin Layout

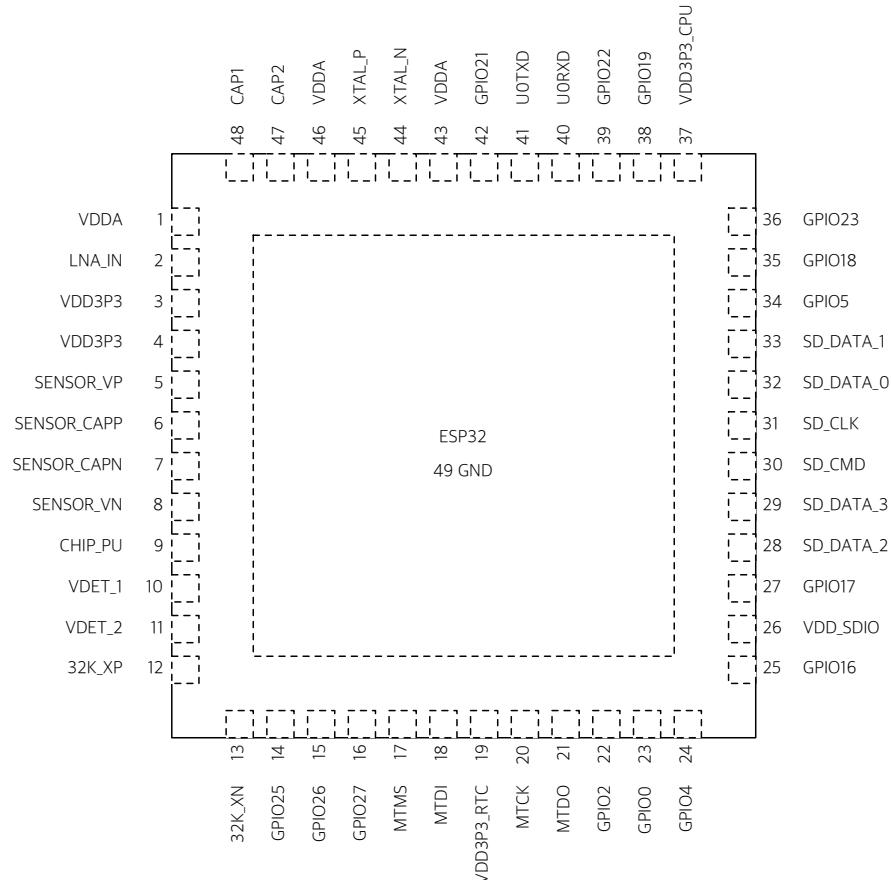


Figure 2: ESP32 Pin Layout (for QFN 6*6)

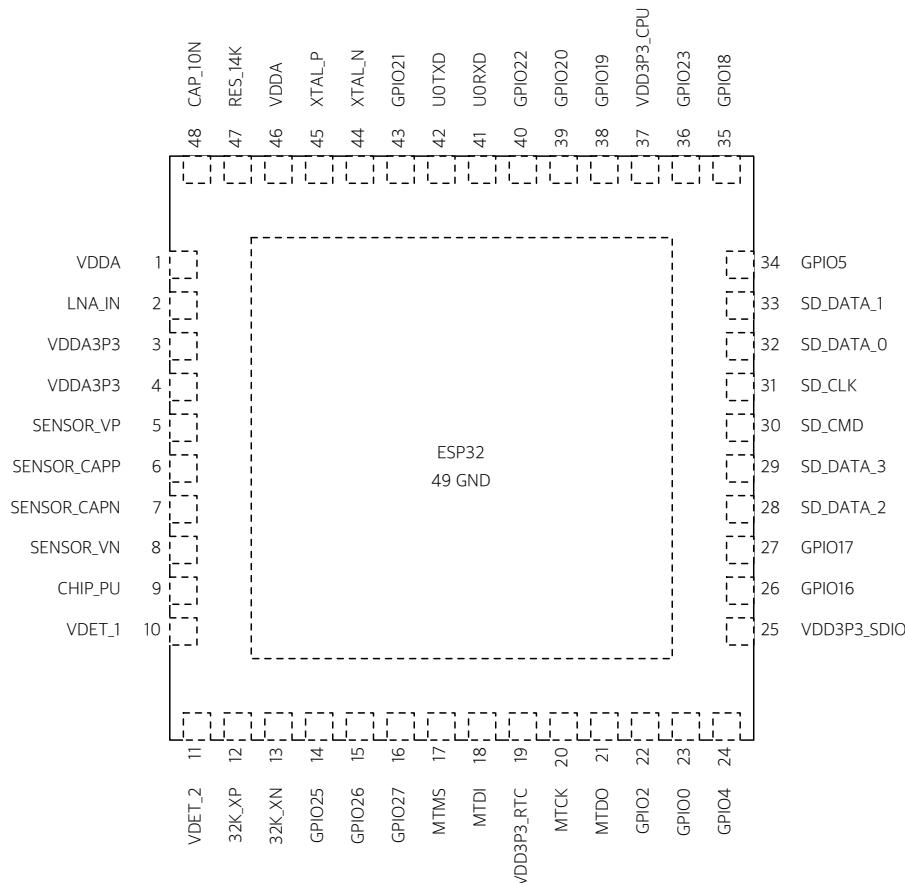


Figure 3: ESP32 Pin Layout (for QFN 5*5)

Note:

For details on ESP32's part number and the corresponding packaging information, please refer to [ESP32 Datasheet](#).

2.2 Pin Description

Table 1: Pin Description

Name	No.	Type	Function
Analog			
VDDA	1	P	Analog power supply (2.3V ~ 3.6V)
LNA_IN	2	I/O	RF input and output
VDD3P3	3	P	Amplifier power supply (2.3V ~ 3.6V)
VDD3P3	4	P	Amplifier power supply (2.3V ~ 3.6V)
VDD3P3_RTC			
SENSOR_VP	5	I	GPIO36, ADC_PRE_AMP, ADC1_CH0, RTC_GPIO0 Note: Connects 270 pF capacitor from SENSOR_VP to SENSOR_CAPP when used as ADC_PRE_AMP.

Name	No.	Type	Function
SENSOR_CAPP	6	I	GPIO37, ADC_PRE_AMP, ADC1_CH1, RTC_GPIO1 Note: Connects 270 pF capacitor from SENSOR_VP to SENSOR_CAPP when used as ADC_PRE_AMP.
SENSOR_CAPN	7	I	GPIO38, ADC1_CH2, ADC_PRE_AMP, RTC_GPIO2 Note: Connects 270 pF capacitor from SENSOR_VN to SENSOR_CAPN when used as ADC_PRE_AMP.
SENSOR_VN	8	I	GPIO39, ADC1_CH3, ADC_PRE_AMP, RTC_GPIO3 Note: Connects 270 pF capacitor from SENSOR_VN to SENSOR_CAPN when used as ADC_PRE_AMP.
CHIP_PU	9	I	Chip Enable (Active High) High: On, chip works normally Low: Off, chip works at the minimum power Note: Do not leave CHIP_PU pin floating
VDET_1	10	I	GPIO34, ADC1_CH6, RTC_GPIO4
VDET_2	11	I	GPIO35, ADC1_CH7, RTC_GPIO5
32K_XP	12	I/O	GPIO32, 32K_XP (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
32K_XN	13	I/O	GPIO33, 32K_XN (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8
GPIO25	14	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
GPIO26	15	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
GPIO27	16	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
MTMS	17	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPI-CLK, HS2_CLK, SD_CLK, EMAC_TXD2
MTDI	18	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
VDD3P3_RTC	19	P	RTC IO power supply input (1.8V ~ 3.3V)
MTCK	20	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
MTDO	21	I/O	GPIO15, ADC2_CH3, TOUCH3, RTC_GPIO13, MTDO, HSPICS0, HS2_CMD, SD_CMD, EMAC_RXD3
GPIO2	22	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
GPIO0	23	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
GPIO4	24	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
VDD_SDIO			
GPIO16	25	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
VDD_SDIO	26	P	1.8V or 3.3V power supply output
GPIO17	27	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
SD_DATA_2	28	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SD_DATA_3	29	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SD_CMD	30	I/O	GPIO11, SD_CMD, SPICSO, HS1_CMD, U1RTS
SD_CLK	31	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS

Name	No.	Type	Function
SD_DATA_0	32	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SD_DATA_1	33	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
VDD3P3_CPU			
GPIO5	34	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
GPIO18	35	I/O	GPIO18, VSPICLK, HS1_DATA7
GPIO23	36	I/O	GPIO23, VSPID, HS1_STROBE
VDD3P3_CPU	37	P	CPU IO power supply input (1.8V ~ 3.3V)
GPIO19	38	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
GPIO22	39	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
U0RXD	40	I/O	GPIO3, U0RXD, CLK_OUT2
U0TXD	41	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
GPIO21	42	I/O	GPIO21, VSPIHD, EMAC_TX_EN
Analog			
VDDA	43	P	Analog power supply (2.3V ~ 3.6V)
XTAL_N	44	O	External crystal output
XTAL_P	45	I	External crystal input
VDDA	46	P	Power supply for PLL (2.3V ~ 3.6V)
CAP2	47	I	Connects with a 3 nF capacitor and 20 kΩ resistor in parallel to CAP1
CAP1	48	I	Connects with a 10 nF series capacitor to ground
GND			
GND	49	P	GND

Notice:

GPIO36, GPIO37, GPIO38, GPIO39, GPIO34 and GPIO35 can only be used for input.

2.3 Strapping Pins

ESP32 has six strapping pins:

- MTDI/GPIO12: internal pull-down
- GPIO0: internal pull-up
- GPIO2: internal pull-down
- GPIO4: internal pull-down
- MTDO/GPIO15: internal pull-up
- GPIO5: internal pull-up

Software can read the value of these six bits from the register "GPIO_STRAPPING".

During the chip power-on reset, the latches of the strapping pins sample the voltage level as strapping bits with states "0" or "1", and hold these bits until the chip is powered down or shut down. The strapping bits configure the device boot mode, the operating voltage of VDD_SDIO and other system initial settings.

Each strapping pin is connected with its internal pull-up/pull-down during the chip reset. Consequently, if a strapping pin is unconnected or the connected external circuit is high-impedance, the internal weak pull-up/pull-down will determine the default input level of the strapping pins.

To change the strapping bit values and consequently change the ESP32 boot mode, users can apply the external pull-down/pull-up resistances, or use the host MCU's GPIOs to control the voltage level of these pins when powering on the ESP32.

After reset procedure is complete, the strapping pins work as the normal function pins.

Please see Table 2 for detailed information on boot mode configuration using strapping pins.

Table 2: Strapping Pins

Voltage of Internal LDO (VDD_SDIO)					
Pin	Default	3.3V	1.8V		
MTDI	Pull-down	0	1		
Booting Mode					
Pin	Default	SPI Boot	Download Boot		
GPIO0	Pull-up	1	0		
GPIO2	Pull-down	Don't-care	0		
Debugging Log on U0TXD During Booting					
Pin	Default	U0TXD Toggling	U0TXD Silent		
MTDO	Pull-up	1	0		
Timing of SDIO Slave					
Pin	Default	Falling-edge Input Falling-edge Output	Falling-edge Input Rising-edge Output	Rising-edge Input Falling-edge Output	Rising-edge Input Rising-edge Output
MTDO	Pull-up	0	0	1	1
GPIO5	Pull-up	0	1	0	1

Note:

Firmware can configure register bits to change the setting of "Voltage of Internal LDO (VDD_SDIO)" and "Timing of SDIO Slave" after booting.

3. Schematics and PCB Layout Design

ESP32's integrated circuitry requires only 16 resistors, capacitors and sensors, 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.

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.

3.1 Schematics

ESP32 schematics is as shown in Figure 4.

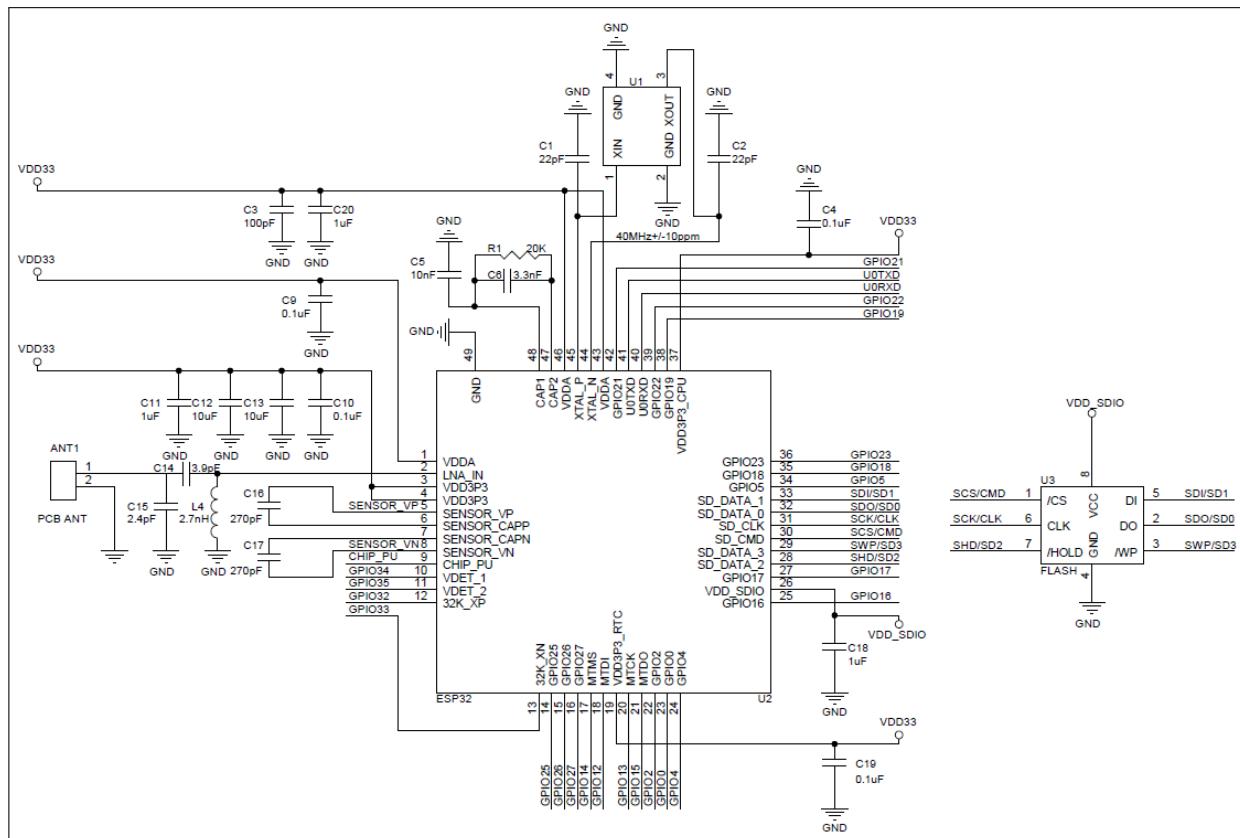


Figure 4: ESP32 Schematics

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

- Sensor external sampling capacitor
- External capacitors

A detailed description of these aspects follows.

3.1.1 Power Supply

3.1.1.1 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 ~ 3.3V. We recommend adding extra filter capacitors close to the digital power supply pins.

The internal LDO of VDD_SDIO can be used as the 1.8V or 3.3V power supply for external circuitry, with a maximum current of about 40 mA. The user can add a $1\mu\text{F}$ filter capacitor close to VDD_SDIO. When VDD_SDIO is tied to VDD3P3_RTC, the LDO will be disabled.

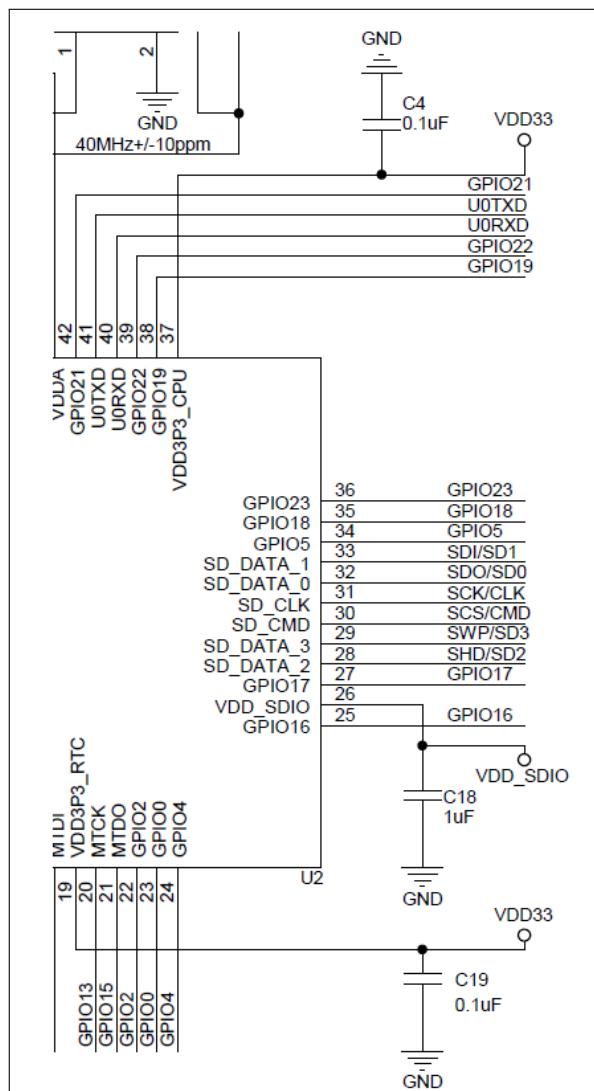


Figure 5: ESP32 Digital Power Supply Pins

3.1.1.2 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 0603 10 μ F capacitor to the power trace, which can work in conjunction with the 0402 0.1 μ F capacitor.

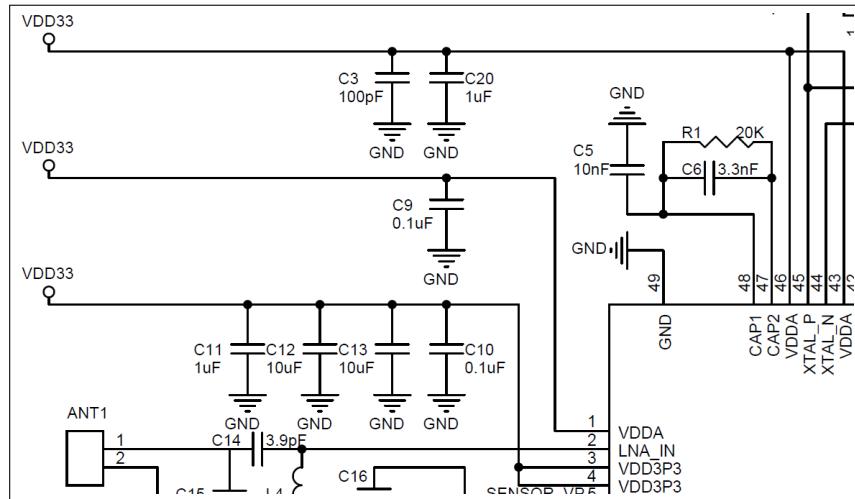


Figure 6: ESP32 Analog Power Supply Pins

Notice:

The operating voltage for ESP32 ranges from 2.3V to 3.6V. When using a single power supply, the recommended voltage of the power supply is 3.3V, and its recommended output current is 500 mA or more.

3.1.2 Power-on Sequence and System Reset

3.1.2.1 Power-on Sequence

ESP32 uses a 3.3V system power supply. There is only one important design practice that should be followed with regard to the power-on sequence: The Pin9 CHIP_PU should be powered on later than, or at the same time as, the 3.3V system power supply pin.

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 ($R=1\text{ k}\Omega$, $C=100\text{ nF}$) is required.

3.1.2.2 Reset

CHIP_PU serves as the reset pin of ESP32. ESP32 will power off when CHIP_PU is held low. To avoid reboots caused by external interferences, the CHIP_PU trace should be as short as possible and routed away from clock

lines. A pull-up resistor and a ground capacitor are highly recommended.

Notice:

CHIP_PU pin must not be left floating.

3.1.3 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 SOIC-8 (SOP-8) 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 0402 serial resistor to Pin21 SD_CLK and connect it to the Flash CLK pin. The resistor can reduce drive current, thus minimizing crosstalk and external interference. The resistor may also be used to tweak the bus timing and sequence.

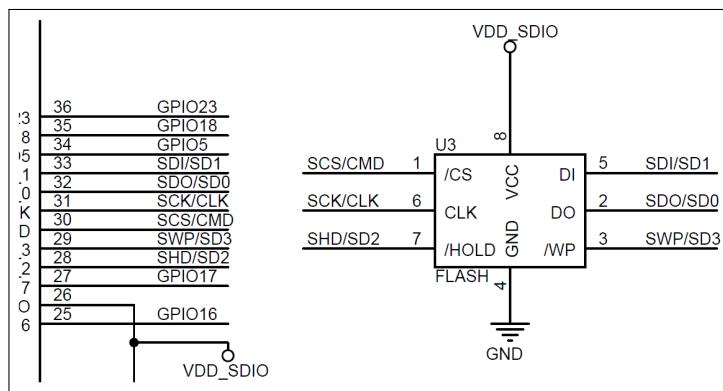


Figure 7: ESP32 Flash

3.1.4 Crystal Oscillator

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

3.1.4.1 External Clock Source (compulsory)

ESP32 can support 40 MHz, 26 MHz, and 24 MHz crystal oscillators. When using ESP32, please make sure you select the right crystal oscillator frequency in the flash download tool. In circuit design, capacitors C1 and C2 are added to the input and output terminals of the crystal oscillator, respectively. The values of the two capacitors may vary based on layout and operating conditions, ranging from 6 pF to 22 pF. However, the exact capacitor values depend on further testing of, and adjustment to, the overall performance of the whole circuit. Normally, the capacitive values of C1 and C2 are within 10 pF for the 26 MHz crystal oscillator. For 40 MHz crystal oscillator, $C1 \geq 10\text{pF}$, $C2 \leq 22\text{pF}$. The accuracy of the crystal oscillator should be ± 10 PPM. The operating temperature should range from -20°C to 85°C.

Notice:

Defects in the craftsmanship of the crystal oscillators (for example, high frequency deviation) and unstable working temperature may lead to the malfunction of ESP32, resulting in the decrease of the overall performance.

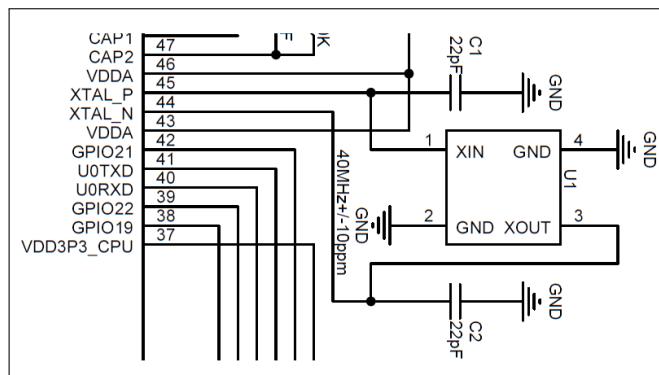


Figure 8: ESP32 Crystal Oscillator

3.1.4.2 RTC (Optional)

ESP32 supports an external 32 kHz crystal oscillator to act as the RTC sleep clock.

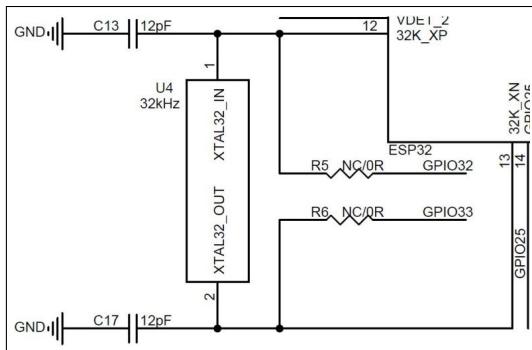


Figure 9: ESP32 Crystal Oscillator (RTC)

Notice:

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

3.1.5 RF

In the circuit design, a π -type matching network is essential for antenna matching.

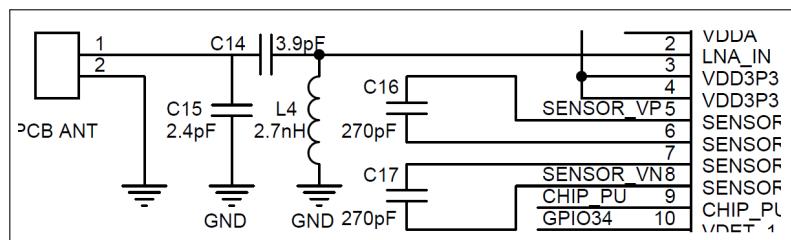


Figure 10: ESP32 RF Matching Schematics

Note:

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

3.1.6 Sensor External Sampling Capacitor

The capacitors (270 pF) between SENSOR_VP and SENSOR_CAPP, SENSOR_CAPN and SENSOR_VN are used as the sampling capacitors for the internal switch amplifier. If the two capacitors are removed, SENSOR_VP, SENSOR_CAPP, SENSOR_CAPN and SENSOR_VN can be used as normal ADCs.

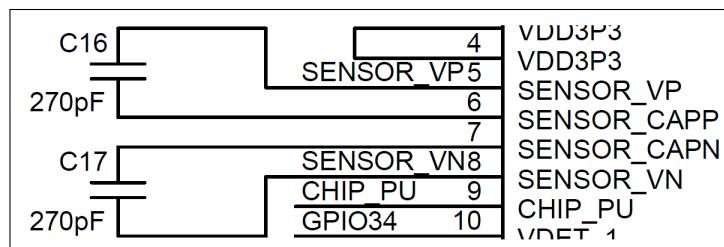


Figure 11: ESP32 Sensor External Sampling Capacitor

3.1.7 External Capacitor

The schematics of Pin47 CAP2 and Pin48 CAP1 is shown in Figure 12. 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 12. Removing the RC circuit may slightly affect ESP32 in Deep-sleep mode.

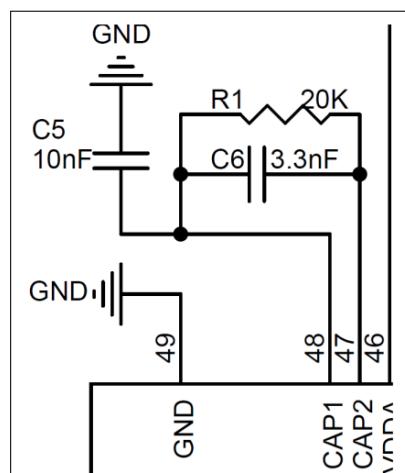


Figure 12: ESP32 External Capacitor

3.2 PCB Layout

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.

3.2.1 Standalone ESP32 Module

3.2.1.1 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.

3.2.1.2 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 0603 10 μF capacitor and a 0402 0.1 μF capacitor are required. As Figure 13 shows, C13 (10 μF capacitor) is placed by the 3.3V stamp hole, and C10 is placed as close as possible to the analog power-supply pin.

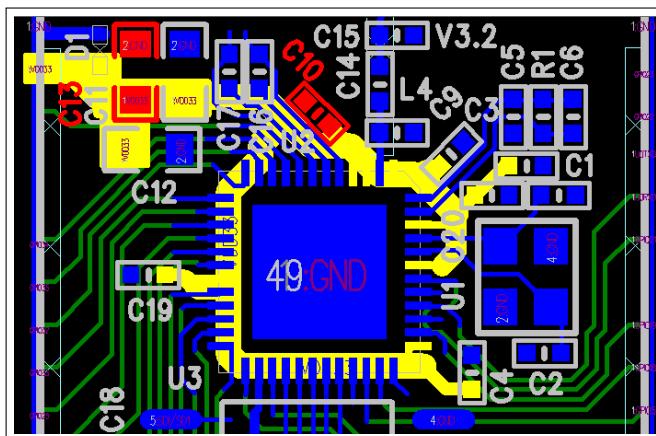


Figure 13: ESP32 PCB Layout

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.

3.2.1.3 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.

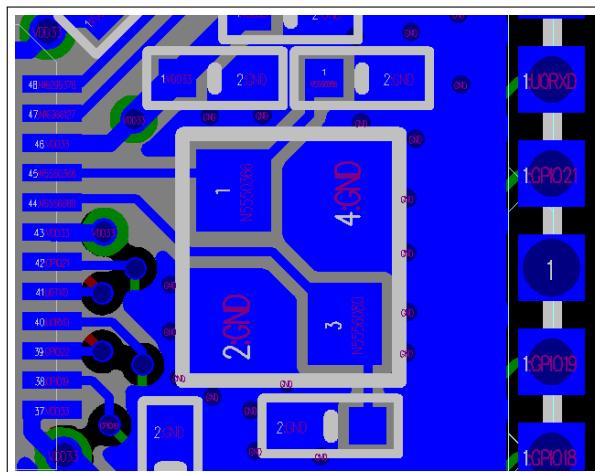


Figure 14: ESP32 Crystal Oscillator Layout

3.2.1.4 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.

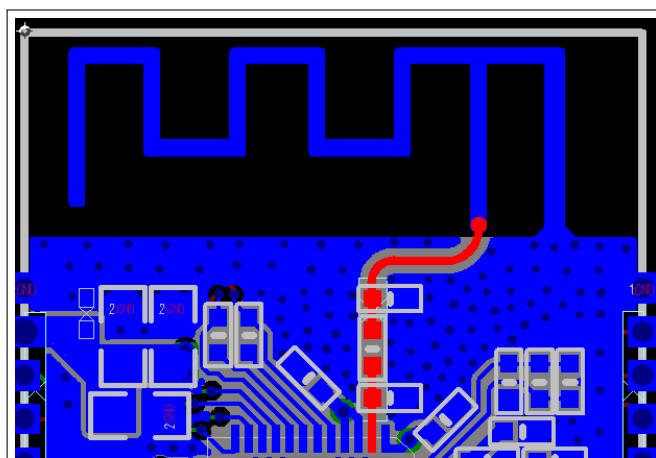


Figure 15: ESP32 RF Layout

3.2.2 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.

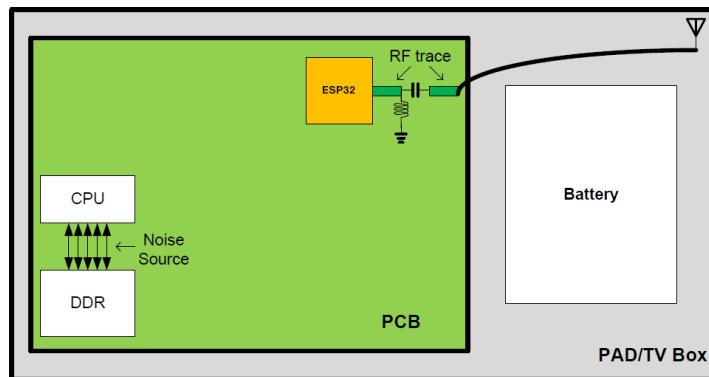


Figure 16: PAD/TV Box Layout

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 16, 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 200Ω series resistor is added to the six signal traces when ESP32 communicates with the CPU via SDIO to decrease the drive current and any interferences, 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.

3.2.3 Typical Layout Problems and Solutions

3.2.3.1 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 ripple should be <100 mV when ESP32 sends 11n MCS7 packets, and < 120 mV when ESP32 sends 11b 11m packets.

Solution:

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

3.2.3.2 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 [3.2](#) for details.

3.2.3.3 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.

3.2.3.4 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 [3.2](#) for details.

4. Hardware Development

4.1 ESP-WROOM-32 Module

4.1.1 Overview

Espressif provides users with an SMD module, the ESP-WROOM-32. At the core of this module is the ESP32-D0WDQ6 chip*. This module has been adjusted, in order to achieve the optimum RF performance.

Note:

* For details on the part number of the ESP32 series, please refer to the document [ESP32 Datasheet](#).

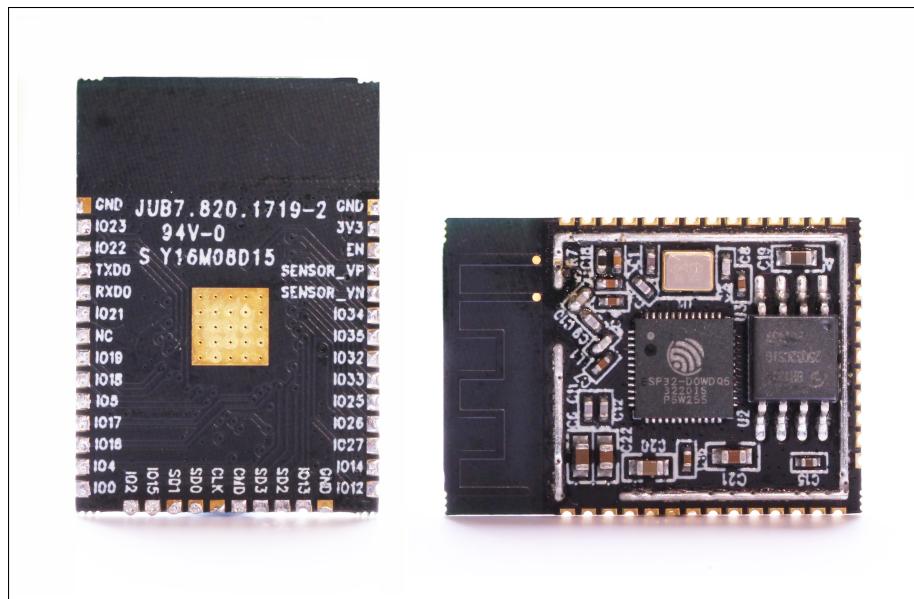


Figure 17: ESP-WROOM-32 Module

The size of the module is 18 mm x 25.5 mm. The flash used is in an SOP8-208 mil package. The on-board PCB antenna has a gain of 2 dBi. Figure 18 shows the dimensions and pin layout of this module.

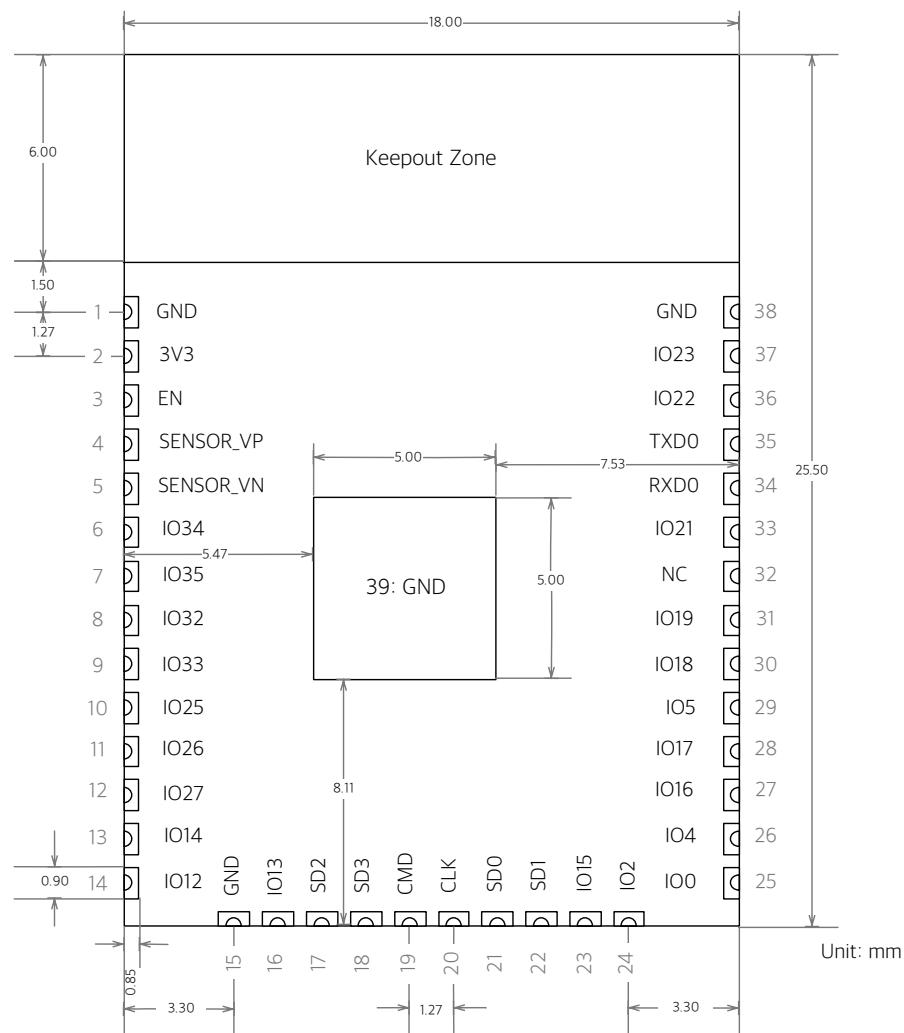


Figure 18: ESP-WROOM-32 Pin Layout

4.1.2 Pin Definition

There are 39 pins led out onto the module. The pin descriptions are listed in Table 3 below:

Table 3: ESP-WROOM-32 Pin Definitions

Name	No.	Type	Function
GND	1	P	Ground
3V3	2	P	Power supply.
EN	3	I	Chip-enable signal. Active high.
SENSOR_VP	4	I	GPIO36, SENSOR_VP, ADC_H, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	I	GPIO39, SENSOR_VN, ADC1_CH3, ADC_H, RTC_GPIO3
IO34	6	I	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	I	GPIO35, ADC1_CH7, RTC_GPIO5
IO32	8	I/O	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	I/O	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8
IO25	10	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0

Name	No.	Type	Function
IO26	11	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
IO14	13	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12	14	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	P	Ground
IO13	16	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPIID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
SHD/SD2	17	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SWP/SD3	18	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SCS/CMD	19	I/O	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
SCK/CLK	20	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
SDO/SD0	21	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SDI/SD1	22	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
IO15	23	I/O	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
IO2	24	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
IO0	25	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
IO4	26	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
IO16	27	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
IO17	28	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
IO5	29	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
IO18	30	I/O	GPIO18, VSPICLK, HS1_DATA7
IO19	31	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
NC	32	-	-
IO21	33	I/O	GPIO21, VSPIHD, EMAC_TX_EN
RXD0	34	I/O	GPIO3, U0RXD, CLK_OUT2
TXD0	35	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
IO22	36	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
IO23	37	I/O	GPIO23, VSPIID, HS1_STROBE
GND	38	P	Ground
GND	39	P	Ground

4.1.3 Notes

- The module uses one single pin as the power supply pin. The user 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 Wi-Fi functionality. Set the EN pin high for normal working mode.
- 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 this time when the module is powered on. Once the module is powered down, the internal memory will clear up. However, if the firmware is burnt into the flash, it will be stored and can 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.
2. Pull IO0 and IO2 low.
3. Power on the module and check through the serial terminal if the UART Download mode is enabled.
4. Download the firmware to flash using the ESP Flash Download Tool.
5. After downloading, pull IO0 high to enable the SPI Boot mode.
6. 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 opened for both the log-print and flash-download tools simultaneously.
- Please download the ESP Flash Download Tool from Espressif's website: [Flash Download Tool](#).

4.2 ESP32-DevKitC

4.2.1 Overview

ESP32-DevKitC is a low-footprint, minimal system development board which is powered by our latest ESP-WROOM-32. The dimensions of the board are shown in Figure 19.

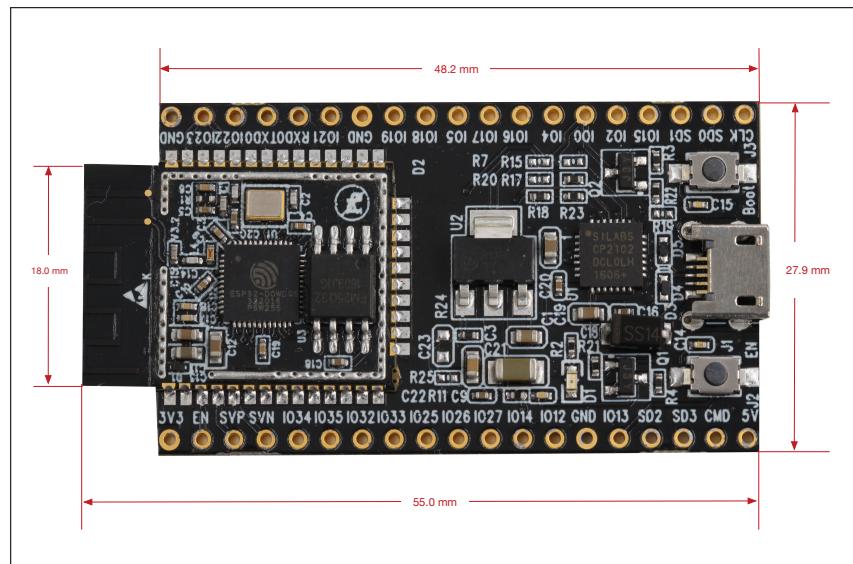


Figure 19: Top view of ESP32-DevKitC

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. The interfaces are shown in Figure 20. For more details, please see Section 4.1.2 ESP-WROOM-32 Pin Definition.

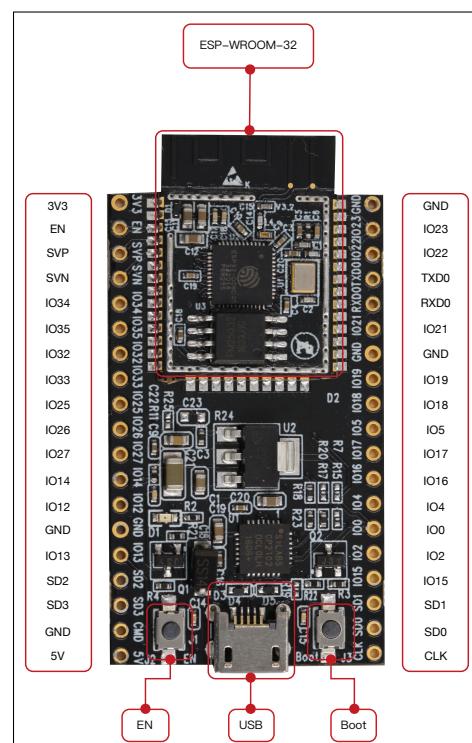


Figure 20: ESP32-DevKitC Pin Layout

4.2.2 Schematics

4.2.2.1 Power Schematics

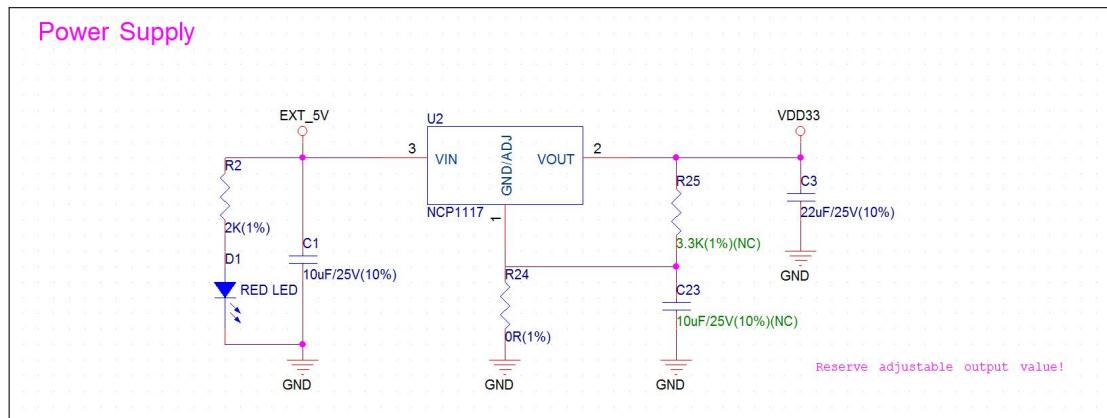


Figure 21: Power Schematics of ESP32-DevKitC

4.2.2.2 USB-UART Schematics

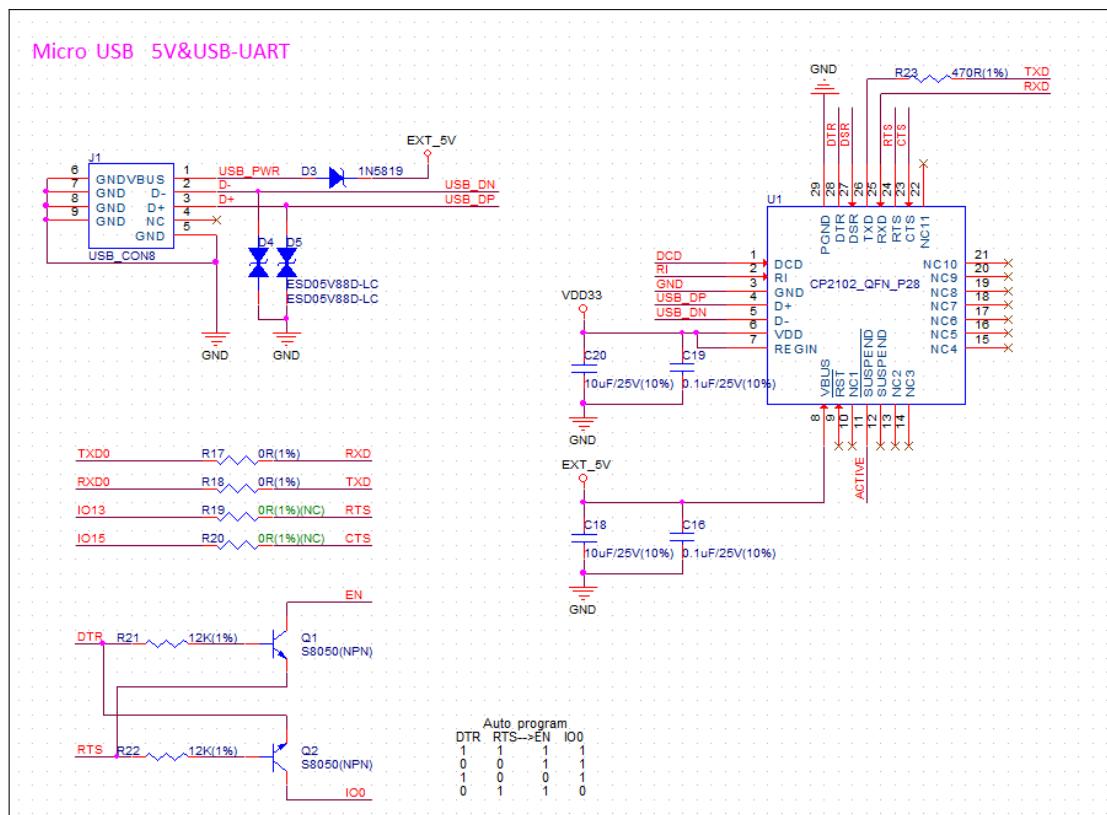


Figure 22: USB-UART Schematics

5. Applications

5.1 UART to Wi-Fi Smart Device

The two UART interfaces are defined in Table 4.

Table 4: 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 4.2 ESP32-DevKitC).

5.2 ESP32-Lyra Smart Audio Platform

ESP32-Lyra is a cost-effective smart audio platform, which is specifically designed by Espressif for the IoT industry. With its ESP32 dual-core processor and Wi-Fi + BT capability, ESP32-Lyra features voice recognition, audio playing, and access to cloud services. The ESP32-Lyra platform supports systems of artificial intelligence, voice and image recognition, wireless audio systems, as well as smart home networks.

The ESP32-Lyra Smart Audio Platform has the following features:

- Support for multiple audio interfaces with high extensibility
- Support for touch buttons
- Support for multiple audio formats including WMA, ALAC, AAC, FLAC, OPUS, MP3, WAV, and OGG
- Support for multiple wireless audio standards including DLNA, AirPlay and QPlay
- Support for multiple cloud platforms including Ximalaya FM, YunOS and Amazon
- Support for multiple distribution network protocols including ESP-TOUCH, ALINK, JoyLink3.0 and AirKiss