Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!


## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

# ESP32 Datasheet

Espressif Systems

April 11, 2017

# About This Guide

This document provides introduction to the specifications of ESP32 hardware.

The document structure is as follows:

| Chapter | Title | Subject |
|---|---|---|
| Chapter 1 | Overview | An overview of ESP32, including featured solutions, basic and advanced features, applications and development support. |
| Chapter 2 | Pin Definitions | Introduction to the pin layout and descriptions. |
| Chapter 3 | Functional Description | Description of the major functional modules. |
| Chapter 4 | Peripheral Interface | Description of the peripheral interfaces integrated on ESP32. |
| Chapter 5 | Electrical Characteristics | The electrical characteristics and data of ESP32. |
| Chapter 6 | Package Information | The package details of ESP32. |
| Chapter 7 | Part Number and Ordering Information | The part number and ordering information of the ESP32 series. |
| Chapter 8 | Supported Resources | The ESP32-related documents and community resources. |
| Appendix A | Touch Sensor | The touch sensor design and layout guidelines. |
| Appendix B | Code Examples | Input and output code examples. |
| Appendix C | ESP32 Pin Lists | Lists of ESP32's GPIO_Matrix, Ethernet_MAC and IO_MUX pins. |

## Release Notes

| Date | Version | Release notes |
|---|---|---|
| 2016.08 | V1.0 | First release. |
| 2017.02 | V1.1 | Added Chapter Part Number and Ordering Information;<br>Updated Section MCU and Advanced Features;<br>Updated Section Block Diagram;<br>Updated Chapter Pin Definitions;<br>Updated Section CPU and Memory;<br>Updated Section Audio PLL Clock;<br>Updated Section Recommended Operating Conditions;<br>Updated Chapter Package Information;<br>Updated Chapter Learning Resources. |
| 2017.03 | V1.2 | Added a note to Table Pin Description;<br>Updated the note in Section Internal Memory. |
| 2017.04 | V1.3 | Added Appendix ESP32 Pin Lists;<br>Updated Table Wi-Fi Radio Characteristics;<br>Updated Figure ESP32 Pin Layout (for QFN 5*5). |

## 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 performance and RF performance, showing robustness, versatility, excellent features and reliability in a wide variety of applications and different power profiles.

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 Part Number and Ordering Information.

## 1.1 Featured Solutions

### 1.1.1 Ultra-Low-Power-Solution

ESP32 is designed for mobile, wearable electronics, and Internet of Things (IoT) applications. It has many features of the state-of-the-art low power chips, including fine resolution clock gating, power modes, and dynamic power scaling.

For instance, in a low-power IoT sensor hub application scenario, ESP32 is woken up periodically and only when a specified condition is detected; low duty cycle is used to minimize the amount of energy that the chip expends. The output power of the power amplifier is also adjustable to achieve an optimal trade-off between communication range, data rate and power consumption.

**Note:**
For more information, refer to Section 3.7 RTC and Low-Power Management.

### 1.1.2 Complete Integration Solution

ESP32 is the most integrated solution for Wi-Fi + Bluetooth applications in the industry with less than 10 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.

## 1.2 Basic Protocols

### 1.2.1 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

**Note:**
For more information, refer to Section 3.5 Wi-Fi.

### 1.2.2 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

## 1.3 MCU and Advanced Features

### 1.3.1 CPU and Memory

- Xtensa® Single-/Dual-Core 32-bit LX6 microprocessor(s), up to 600 DMIPS
- 448 KB ROM
- 520 KB SRAM
- 16 KB SRAM in RTC
- QSPI flash/SRAM, up to 4 x 16 MB
- Power supply: 2.2V to 3.6V

### 1.3.2 Clocks and Timers

- Internal 8 MHz oscillator with calibration
- Internal RC oscillator with calibration
- External 2 MHz to 40 MHz crystal oscillator
- External 32 kHz crystal oscillator for RTC with calibration
- Two timer groups, including 2 x 64-bit timers and 1 x main watchdog in each group
- RTC timer with sub-second accuracy
- RTC watchdog

### 1.3.3 Advanced Peripheral Interfaces

- 12-bit SAR ADC up to 18 channels
- 2 × 8-bit D/A converters
- 10 × touch sensors
- Temperature sensor
- 4 × SPI
- 2 × I2S
- 2 × I2C
- 3 × UART
- 1 host (SD/eMMC/SDIO)
- 1 slave (SDIO/SPI)
- Ethernet MAC interface with dedicated DMA and IEEE 1588 support
- CAN 2.0
- IR (TX/RX)
- Motor PWM
- LED PWM up to 16 channels
- Hall sensor
- Ultra-low-noise analog pre-amplifier

### 1.3.4 Security

- IEEE 802.11 standard security features all supported, including WFA, WPA/WPA2 and WAPI
- Secure boot
- Flash encryption
- 1024-bit OTP, up to 768-bit for customers
- Cryptographic hardware acceleration:
  - AES
  - HASH (SHA-2) library
  - RSA
  - ECC
  - Random Number Generator (RNG)

### 1.3.5 Development Support

- SDK Firmware for fast on-line programming
- Open source toolchains based on GCC

**Note:**
For more information, please refer to Learnig Resources.

## 1.4 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

## 1.5 Block Diagram

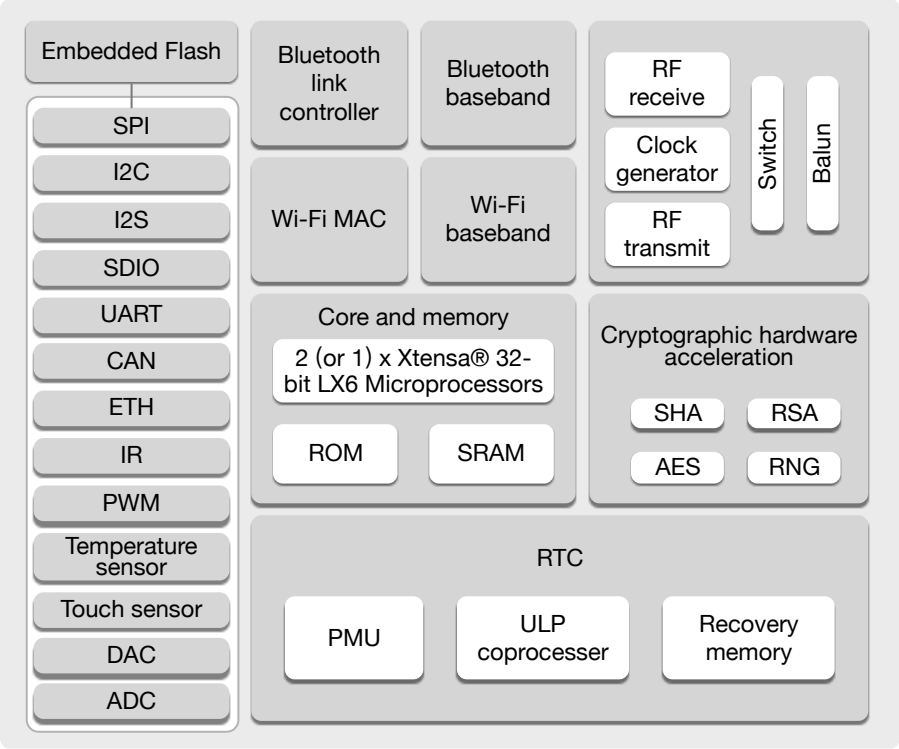


**Figure 1: Function Block Diagram**

**Note:**
Products in the ESP32 series differ from each other in terms of their support for embedded flash and the number of CPUs they have. For details, please refer to Part Number and Ordering Information.

# 2. Pin Definitions

## 2.1 Pin Layout

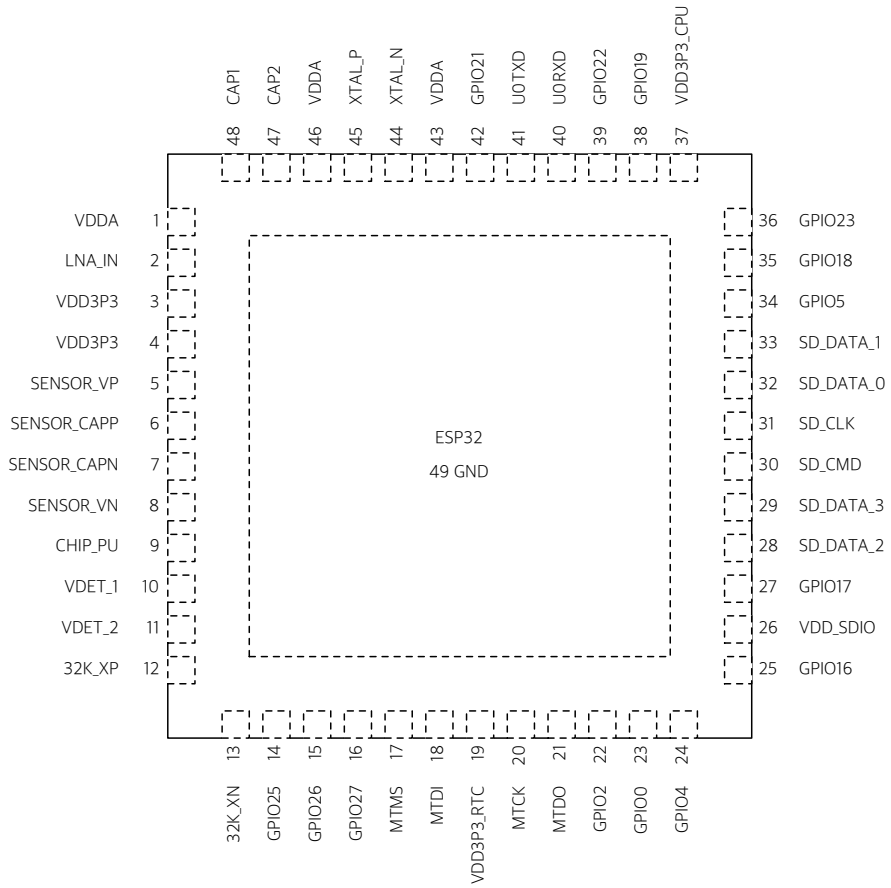


**Figure 2: ESP32 Pin Layout (for QFN 6*6)**

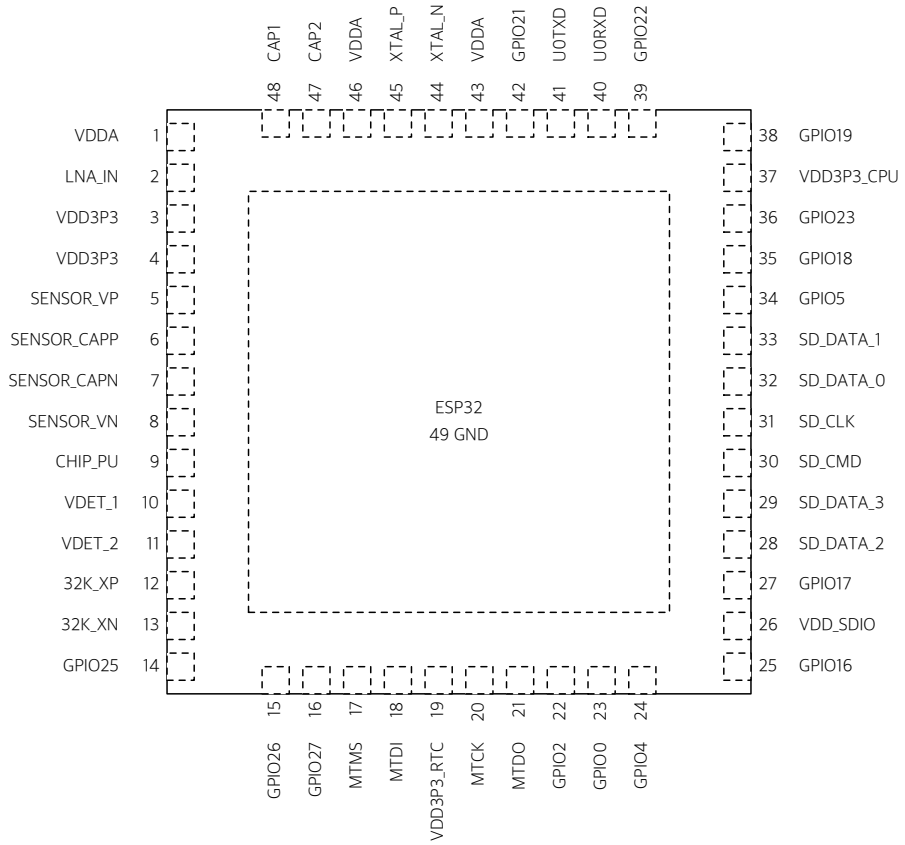


**Figure 3: ESP32 Pin Layout (for QFN 5*5)**

**Note:**
For details on ESP32's part number and the corresponding packaging, please refer to Part Number and Ordering Information.

## 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<br>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<br>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<br>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<br>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)<br>High: On, chip works properly<br>Low: Off, chip works at the minimum power<br>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, HSPICLK, 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, SPICS0, 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 | Digital 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 | 49 | P | Ground |

**Note:**

- ESP32-D2WD's pins GPIO16, GPIO17, SD_CMD, SD_CLK, SD_DATA_0 and SD_DATA_1 are used for connecting the embedded flash, and are not recommended for other uses.
- For a quick reference guide of the IO MUX, Ethernet MAC, and GIPO Matrix pins of ESP32, please refer to Appendix C: ESP32 Pin Lists.

## 2.3 Power Scheme

ESP32 digital pins are divided into three different power domains:

- VDD3P3_RTC
- VDD3P3_CPU
- VDD_SDIO

**VDD3P3_RTC** is also the input power supply for RTC and CPU. **VDD3P3_CPU** is also the input power supply for CPU.
**VDD_SDIO** connects to the output of an internal LDO, whose input is **VDD3P3_RTC**. When **VDD_SDIO** is connected to the same PCB net together with **VDD3P3_RTC**; the internal LDO is disabled automatically.

The internal LDO can be configured as 1.8V, or the same voltage as **VDD3P3_RTC**. It can be powered off via software to minimize the current of flash/SRAM during the Deep-sleep mode.

**Note:**

- It is required that the power supply of **VDD3P3_RTC**, **VDD3P3_CPU** and analog must be stable before the pin **CHIP_PU** is set at high level.
- 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.

## 2.4 Strapping Pins

ESP32 has five strapping pins:

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

Software can read the value of these five 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 of "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-impendence, the internal weak pull-up/pull-down will determine the default input level of the strapping pins.

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

After reset, the strapping pins work as the normal functions pins.

Refer to Table 2 for detailed boot modes configuration by 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. Functional Description

This chapter describes the functions integrated in ESP32.

## 3.1 CPU and Memory

### 3.1.1 CPU

ESP32 contains one/two low-power Xtensa® 32-bit LX6 microprocessor(s) with the following features.

- 7-stage pipeline to support the clock frequency of up to 240 MHz
- 16/24-bit Instruction Set provides high code-density
- Support Floating Point Unit
- Support DSP instructions, such as 32-bit Multiplier, 32-bit Divider, and 40-bit MAC
- Support 32 interrupt vectors from about 70 interrupt sources

The single-/dual-CPU interfaces include:

- Xtensa RAM/ROM Interface for instruction and data
- Xtensa Local Memory Interface for fast peripheral register access
- Interrupt with external and internal sources
- JTAG interface for debugging

### 3.1.2 Internal Memory

ESP32's internal memory includes:

- 448 KB ROM for booting and core functions
- 520 KB on-chip SRAM for data and instruction
- 8 KB SRAM in RTC, which is called RTC SLOW Memory and can be used for co-processor accessing during the Deep-sleep mode
- 8 KB SRAM in RTC, which is called RTC FAST Memory and can be used for data storage and the main CPU during RTC Boot from the Deep-sleep mode
- 1 kbit of eFuse, of which 256 bits are used for the system (MAC address and chip configuration) and the remaining 768 bits are reserved for customer applications, including Flash-Encryption and Chip-ID
- Embedded flash

**Note:**

- Products in the ESP32 series differ from each other in terms of their support for embedded flash and the size of the embedded flash. For details, please refer to Part Number and Ordering Information.
- From the ESP32 series of chips specified in this document, ESP32-D2WD has 16 Mbits of embedded flash, connected via pins GPIO16, GPIO17, SD_CMD, SD_CLK, SD_DATA_0 and SD_DATA_1. The other chips in the ESP32 series have no embedded flash.

### 3.1.3 External Flash and SRAM

ESP32 supports up to four 16-MB external QSPI flash and SRAM with hardware encryption based on AES to protect developer's programs and data.

ESP32 can access the external QSPI flash and SRAM through high-speed caches.

- Up to 16 MB of external flash are memory-mapped onto the CPU code space, supporting 8-bit, 16-bit and 32-bit access. Code execution is supported.
- Up to 8 MB of external flash/SRAM memory are mapped onto the CPU data space, supporting 8-bit, 16-bit and 32-bit access. Data-read is supported on the flash and SRAM. Data-write is supported on the SRAM.

**Note:**
ESP32 chips with embedded flash do not support the address mapping between external flash and peripherals.

### 3.1.4 Memory Map

The structure of address mapping is shown in Figure 4. The memory and peripherals mapping of ESP32 is shown in Table 3.

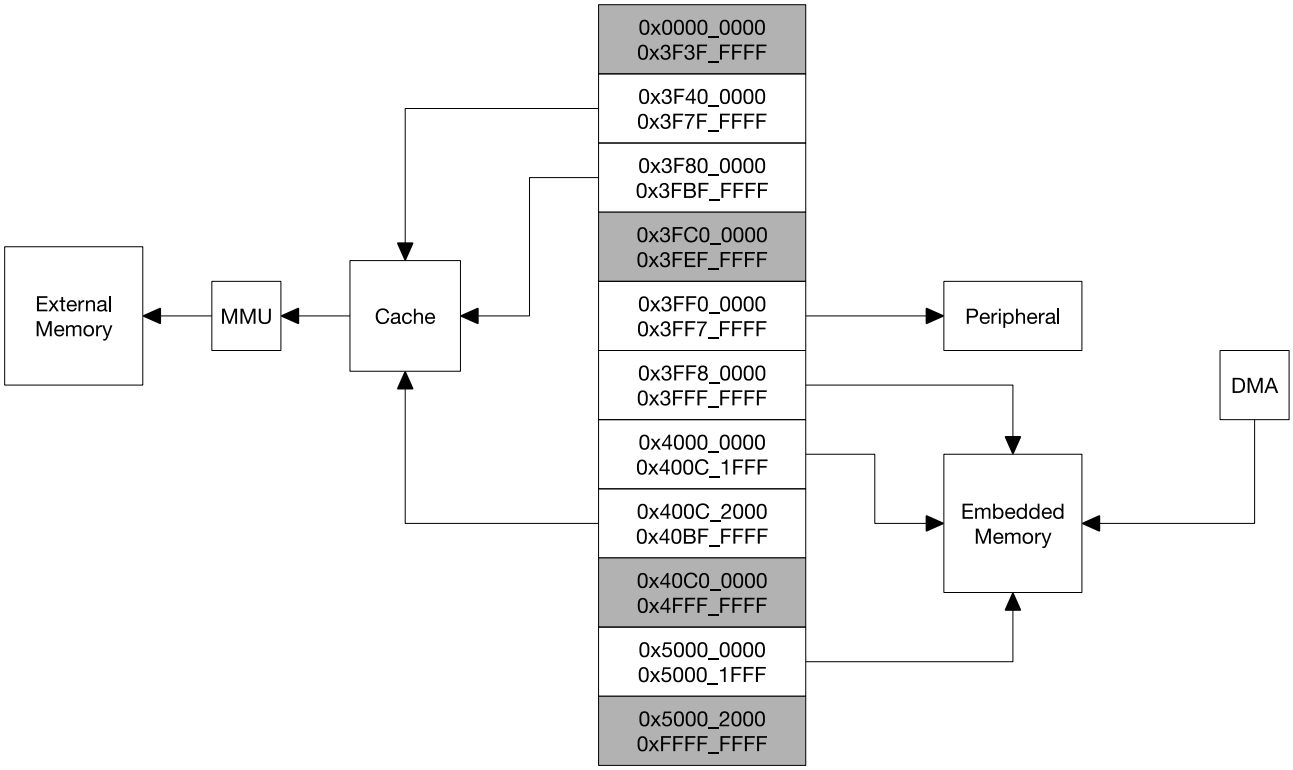


**Figure 4: Address Mapping Structure**

**Table 3: Memory and Peripheral Mapping**

| Category | Target | Start Address | End Address | Size |
|---|---|---|---|---|
| Embedded Memory | Internal ROM 0 | 0x4000_0000 | 0x4005_FFFF | 384 KB |
| | Internal ROM 1 | 0x3FF9_0000 | 0x3FF9_FFFF | 64 KB |
| | Internal SRAM 0 | 0x4007_0000 | 0x4009_FFFF | 192 KB |
| | Internal SRAM 1 | 0x3FFE_0000 | 0x3FFF_FFFF | 128 KB |
| | | 0x400A_0000 | 0x400B_FFFF | |
| | Internal SRAM 2 | 0x3FFA_E000 | 0x3FFD_FFFF | 200 KB |
| | RTC FAST Memory | 0x3FF8_0000 | 0x3FF8_1FFF | 8 KB |
| | | 0x400C_0000 | 0x400C_1FFF | |
| | RTC SLOW Memory | 0x5000_0000 | 0x5000_1FFF | 8 KB |
| External Memory | External Flash | 0x3F40_0000 | 0x3F7F_FFFF | 4 MB |
| | | 0x400C_2000 | 0x40BF_FFFF | 11 MB |
| | | | | 248 KB |
| | External SRAM | 0x3F80_0000 | 0x3FBF_FFFF | 4 MB |
| Peripheral | DPort Register | 0x3FF0_0000 | 0x3FF0_0FFF | 4 KB |
| | AES Accelerator | 0x3FF0_1000 | 0x3FF0_1FFF | 4 KB |
| | RSA Accelerator | 0x3FF0_2000 | 0x3FF0_2FFF | 4 KB |
| | SHA Accelerator | 0x3FF0_3000 | 0x3FF0_3FFF | 4 KB |
| | Secure Boot | 0x3FF0_4000 | 0x3FF0_4FFF | 4 KB |
| | Cache MMU Table | 0x3FF1_0000 | 0x3FF1_3FFF | 16 KB |
| | PID Controller | 0x3FF1_F000 | 0x3FF1_FFFF | 4 KB |
| | UART0 | 0x3FF4_0000 | 0x3FF4_0FFF | 4 KB |
| | SPI1 | 0x3FF4_2000 | 0x3FF4_2FFF | 4 KB |
| | SPI0 | 0x3FF4_3000 | 0x3FF4_3FFF | 4 KB |
| | GPIO | 0x3FF4_4000 | 0x3FF4_4FFF | 4 KB |
| | RTC | 0x3FF4_8000 | 0x3FF4_8FFF | 4 KB |
| | IO MUX | 0x3FF4_9000 | 0x3FF4_9FFF | 4 KB |
| | SDIO Slave | 0x3FF4_B000 | 0x3FF4_BFFF | 4 KB |
| | UDMA1 | 0x3FF4_C000 | 0x3FF4_CFFF | 4 KB |
| | I2S0 | 0x3FF4_F000 | 0x3FF4_FFFF | 4 KB |
| | UART1 | 0x3FF5_0000 | 0x3FF5_0FFF | 4 KB |
| | I2C0 | 0x3FF5_3000 | 0x3FF5_3FFF | 4 KB |
| | UDMA0 | 0x3FF5_4000 | 0x3FF5_4FFF | 4 KB |
| | SDIO Slave | 0x3FF5_5000 | 0x3FF5_5FFF | 4 KB |
| | RMT | 0x3FF5_6000 | 0x3FF5_6FFF | 4 KB |
| | PCNT | 0x3FF5_7000 | 0x3FF5_7FFF | 4 KB |
| | SDIO Slave | 0x3FF5_8000 | 0x3FF5_8FFF | 4 KB |
| | LED PWM | 0x3FF5_9000 | 0x3FF5_9FFF | 4 KB |
| | Efuse Controller | 0x3FF5_A000 | 0x3FF5_AFFF | 4 KB |
| | Flash Encryption | 0x3FF5_B000 | 0x3FF5_BFFF | 4 KB |
| | PWM0 | 0x3FF5_E000 | 0x3FF5_EFFF | 4 KB |
| | TIMG0 | 0x3FF5_F000 | 0x3FF5_FFFF | 4 KB |
| | TIMG1 | 0x3FF6_0000 | 0x3FF6_0FFF | 4 KB |

| Category | Target | Start Address | End Address | Size |
|---|---|---|---|---|
| Peripheral | SPI2 | 0x3FF6_4000 | 0x3FF6_4FFF | 4 KB |
| | SPI3 | 0x3FF6_5000 | 0x3FF6_5FFF | 4 KB |
| | SYSCON | 0x3FF6_6000 | 0x3FF6_6FFF | 4 KB |
| | I2C1 | 0x3FF6_7000 | 0x3FF6_7FFF | 4 KB |
| | SDMMC | 0x3FF6_8000 | 0x3FF6_8FFF | 4 KB |
| | EMAC | 0x3FF6_9000 | 0x3FF6_AFFF | 8 KB |
| | PWM1 | 0x3FF6_C000 | 0x3FF6_CFFF | 4 KB |
| | I2S1 | 0x3FF6_D000 | 0x3FF6_DFFF | 4 KB |
| | UART2 | 0x3FF6_E000 | 0x3FF6_EFFF | 4 KB |
| | PWM2 | 0x3FF6_F000 | 0x3FF6_FFFF | 4 KB |
| | PWM3 | 0x3FF7_0000 | 0x3FF7_0FFF | 4 KB |
| | RNG | 0x3FF7_5000 | 0x3FF7_5FFF | 4 KB |

## 3.2 Timers and Watchdogs

### 3.2.1 64-bit Timers

There are four general-purpose timers embedded in the ESP32. They are all 64-bit generic timers which are based on 16-bit prescalers and 64-bit auto-reload-capable up/downcounters.

The timers feature:

- A 16-bit clock prescaler, from 2 to 65536
- A 64-bit time-base counter
- Configurable up/down time-base counter: incrementing or decrmenting
- Halt and resume of time-base counter
- Auto-reload at alarming
- Software-controlled instant reload
- Level and edge interrupt generation

### 3.2.2 Watchdog Timers

The ESP32 has three watchdog timers: one in each of the two timer modules (called the Main Watchdog Timer, or MWDT) and one in the RTC module (called the RTC Watchdog Timer, or RWDT). These watchdog timers are intended to recover from an unforeseen fault, causing the application program to abandon its normal sequence. A watchdog timer has 4 stages. Each stage may take one of three or four actions upon the expiry of a programmed time period for this stage unless the watchdog is fed or disabled. The actions are: interrupt, CPU reset, and core reset, and system reset. Only the RWDT can trigger the system reset, and is able to reset the entire chip, including the RTC itself. A timeout value can be set for each stage individually.

During flash boot the RWDT and the first MWDT start automatically in order to detect and recover from booting problems.

The ESP32 watchdogs have the following features:

- 4 stages, each of which can be configured or disabled separately

- Programmable time period for each stage
- One of 3 or 4 possible actions (interrupt, CPU reset, core reset, and system reset) upon the expiry of each stage
- 32-bit expiry counter
- Write protection, to prevent the RWDT and MWDT configuration from being inadvertently altered
- SPI flash boot protection
  If the boot process from an SPI flash does not complete within a predetermined time period, the watchdog will reboot the entire system.

## 3.3 System Clocks

### 3.3.1 CPU Clock

Upon reset, an external crystal clock source (2 MHz ~ 60 MHz), is selected as the default CPU clock. The external crystal clock source also connects to a PLL to generate a high frequency clock (typically 160 MHz).

In addition, ESP32 has an internal 8 MHz oscillator. The accuracy of the oscillator is guaranteed by design and is stable within the operating temperatures (with a margin error of 1%). Hence, the application can then select the clock source from the external crystal clock source, the PLL clock or the internal 8 MHz oscillator. The selected clock source drives the CPU clock, directly or after division, depending on the application.

### 3.3.2 RTC Clock

The RTC clock has five possible sources:

- external low speed (32 kHz) crystal clock
- external crystal clock divided by 4
- internal RC oscillator (typically about 150 kHz and adjustable)
- internal 8 MHz oscillator
- internal 31.25 kHz clock (derived from the internal 8 MHz oscillator divided by 256)

When the chip is in the normal power mode and needs faster CPU accessing, the application can choose the external high speed crystal clock divided by 4 or the internal 8 MHz oscillator. When the chip operates in the low power mode, the application chooses the external low speed (32 kHz) crystal clock, the internal RC clock or the internal 31.25 kHz clock.

### 3.3.3 Audio PLL Clock

The audio clock is generated by the ultra-low-noise fractional-N PLL. The output frequency of the audio PLL is programmable, from 16 MHz to 128 MHz, and is given by the following formula:

$$f_{\text{out}} = \frac{f_{\text{xtal}}(sdm2 + \frac{sdm1}{2^8} + \frac{sdm0}{2^{16}} + 4)}{2(odiv + 2)}$$

where $f_{\text{out}}$ is the output frequency, $f_{\text{xtal}}$ is the frequency of the crystal oscillator, and $sdm2$, $sdm1$, $sdm0$ and $odiv$ are all integer values, configurable by registers.

## 3.4 Radio

The ESP32 radio consists of the following main blocks:

- 2.4 GHz receiver
- 2.4 GHz transmitter
- bias and regulators
- balun and transmit-receive switch
- clock generator

### 3.4.1 2.4 GHz Receiver

The 2.4 GHz receiver down-converts the 2.4 GHz RF signal to quadrature baseband signals and converts them to the digital domain with 2 high-resolution, high-speed ADCs. To adapt to varying signal channel conditions, RF filters, Automatic Gain Control (AGC), DC offset cancellation circuits and baseband filters are integrated within ESP32.

### 3.4.2 2.4 GHz Transmitter

The 2.4 GHz transmitter up-converts the quadrature baseband signals to the 2.4 GHz RF signal, and drives the antenna with a high powered Complementary Metal Oxide Semiconductor (CMOS) power amplifier. The use of digital calibration further improves the linearity of the power amplifier, enabling state-of-the-art performance of delivering +20.5 dBm of average power for 802.11b transmission and +17 dBm for 802.11n transmission.
Additional calibrations are integrated to cancel any imperfections of the radio, such as:

- Carrier leakage
- I/Q phase matching
- Baseband nonlinearities
- RF nonlinearities
- Antenna matching

These built-in calibration routines reduce the amount of time and required for product test and render test equipment unnecessary.

### 3.4.3 Clock Generator

The clock generator generates quadrature 2.4 GHz clock signals for the receiver and transmitter. All components of the clock generator are integrated on the chip, including all inductors, varactors, filters, regulators and dividers. The clock generator has built-in calibration and self test circuits. Quadrature clock phases and phase noise are optimized on-chip with patented calibration algorithms to ensure the best performance of the receiver and transmitter.

## 3.5 Wi-Fi

ESP32 implements TCP/IP, full 802.11 b/g/n/e/i WLAN MAC protocol, and Wi-Fi Direct specification. It supports Basic Service Set (BSS) STA and SoftAP operations under the Distributed Control Function (DCF) and P2P group operation compliant with the latest Wi-Fi P2P protocol.