# ESP32 Datasheet

## Espressif Systems

August 4, 2017

# About This Guide

This document provides the specifications for 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 | Learning Resources | The ESP32-related documents and community resources. |
| Appendix A | 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 Absolute Maximum Ratings;<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). |
| 2017.05 | V1.4 | • Added a note to the frequency of external crystal oscillator in Section 1.3.2 Clocks and Timers;<br>• Added a note to Section 2.4 Strapping Pins;<br>• Updated Section 3.7 RTC and Low-Power Management;<br>• Changed the maximum driving capability in Table 8 Absolulte Maximum Ratings from 12 mA to 80 mA;<br>• Changed the input impedance value of 50Ω in Table 10 Wi-Fi Radio Characteristics to output impedance value of 30+j10 Ω;<br>• Added a note to No.8 in Table 18 Notes on ESP32 Pin Lists;<br>• Deleted GPIO20 in Table IO_MUX. |

| Date | Version | Release notes |
|---|---|---|
| 2017.06 | V1.5 | • Changed the power supply range in Section 1.3.1 CPU and Memory;<br>• Updated the note in Section 2.3 Power Scheme;<br>• Updated Table 8 Absolute Maximum Ratings;<br>• Changed the drive strength values of the digital output pins in Note8 in Table 18 Notes on ESP32 Pin Lists;<br>• Added Documentation Change Notification. |
| 2017.06 | V1.6 | Corrected two typos:<br>• Changed the number of external components to 20 in Section 1.1.2;<br>• Changed the number of GPIO pins to 34 in Section 4.1. |
| 2017.08 | V1.7 | • Changed the transmitting power to +12 dBm; the sensitivity of NZIF receiver to -97 dBm in Section 1.2.2;<br>• Added a note to Table 2 Pin Description;<br>• Added 160 MHz clock frequency in section 3.1.1;<br>• Changed the transmitting power from 21 dBm to 20.5 dBm in Section 3.5.1;<br>• Changed the dynamic control range of class-1, class-2 and class-3 transmit output powers to "up to 24 dBm"; and changed the dynamic range of NZIF receiver sensitivity to "over 97 dB" in Section 3.6.1;<br>• Updated Table 6 Power Consumption by Power Modes, and added two notes to it;<br>• Updated sections 4.1, 4.3, 4.11;<br>• Updated Table 8 Absolute Maximum Ratings;<br>• Updated Table 9 RF Power Consumption Specifications, and changed the duty cycle on which the transmitters' measurements are based to 50%.<br>• Updated Table 10 Wi-Fi Radio Characteristics and added a note on "Output impedance" to it;<br>• Updated the parameter "Sensitivity" in Table 11, 13, 15;<br>• Updated the parameters "RF transmit power" and "RF power control range", and added the parameter "Gain control step" in Table 12, 14, 16;<br>• Deleted Chapters Touch Sensor and Code Examples;<br>• Added the link to certification download. |

## Documentation Change Notification

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

## Certifications

**Download the certification(s) for the product here.**

## Disclaimer and Copyright Notice

# Contents


1 Overview 1
1.1 Featured Solutions 1
1.1.1 Ultra-Low-Power Solution 1
1.1.2 Complete Integration Solution 1
1.2 Basic Protocols 1
1.2.1 Wi-Fi 1
1.2.2 Bluetooth 2
1.3 MCU and Advanced Features 3
1.3.1 CPU and Memory 3
1.3.2 Clocks and Timers 3
1.3.3 Advanced Peripheral Interfaces 3
1.3.4 Security 4
1.3.5 Development Support 4
1.4 Applications 4
1.5 Block Diagram 5
2 Pin Definitions 6
2.1 Pin Layout 6
2.2 Pin Description 7
2.3 Power Scheme 9
2.4 Strapping Pins 10
3 Functional Description 12
3.1 CPU and Memory 12
3.1.1 CPU 12
3.1.2 Internal Memory 12
3.1.3 External Flash and SRAM 13
3.1.4 Memory Map 13
3.2 Timers and Watchdogs 15
3.2.1 64-bit Timers 15
3.2.2 Watchdog Timers 15
3.3 System Clocks 16
3.3.1 CPU Clock 16
3.3.2 RTC Clock 16
3.3.3 Audio PLL Clock 16
3.4 Radio 16
3.4.1 2.4 GHz Receiver 17
3.4.2 2.4 GHz Transmitter 17
3.4.3 Clock Generator 17
3.5 Wi-Fi 17
3.5.1 Wi-Fi Radio and Baseband 18
3.5.2 Wi-Fi MAC 18
3.5.3 Wi-Fi Firmware 18
3.5.4 Packet Traffic Arbitration (PTA) 19

3.6 Bluetooth 19
3.6.1 Bluetooth Radio and Baseband 19
3.6.2 Bluetooth Interface 19
3.6.3 Bluetooth Stack 19
3.6.4 Bluetooth Link Controller 19
3.7 RTC and Low-Power Management 20

4 Peripherals and Sensors 23
4.1 General Purpose Input / Output Interface (GPIO) 23
4.2 Analog-to-Digital Converter (ADC) 23
4.3 Ultra-Low-Noise Analog Pre-Amplifier 23
4.4 Hall Sensor 23
4.5 Digital-to-Analog Converter (DAC) 23
4.6 Temperature Sensor 24
4.7 Touch Sensor 24
4.8 Ultra-Lower-Power Coprocessor 24
4.9 Ethernet MAC Interface 24
4.10 SD/SDIO/MMC Host Controller 25
4.11 SDIO/SPI Slave Controller 25
4.12 Universal Asynchronous Receiver Transmitter (UART) 26
4.13 I2C Interface 26
4.14 I2S Interface 26
4.15 Infrared Remote Controller 26
4.16 Pulse Counter 26
4.17 Pulse Width Modulation (PWM) 27
4.18 LED PWM 27
4.19 Serial Peripheral Interface (SPI) 27
4.20 Accelerator 27

5 Electrical Characteristics 28
5.1 Absolute Maximum Ratings 28
5.2 RF Power Consumption Specifications 28
5.3 Wi-Fi Radio 29
5.4 Bluetooth Radio 29
5.4.1 Receiver–Basic Data Rate 29
5.4.2 Transmitter – Basic Data Rate 30
5.4.3 Receiver–Enhanced Data Rate 30
5.4.4 Transmitter–Enhanced Data Rate 31
5.5 Bluetooth LE Radio 32
5.5.1 Receiver 32
5.5.2 Transmitter 32

6 Package Information 33

7 Part Number and Ordering Information 34

8 Learning Resources 35

8.1 Must-Read Documents 35
8.2 Must-Have Resources 35

Appendix A – ESP32 Pin Lists 36
A.1. Notes on ESP32 Pin Lists 36
A.2. GPIO_Matrix 38
A.3. Ethernet_MAC 43
A.4. IO_MUX 43

# List of Tables


| | | |
|---|---|---|
| 2 | Pin Description | 7 |
| 3 | Strapping Pins | 10 |
| 4 | Memory and Peripheral Mapping | 14 |
| 5 | Functionalities Depending on the Power Modes | 21 |
| 6 | Power Consumption by Power Modes | 22 |
| 7 | Capacitive Sensing GPIOs Available on ESP32 | 24 |
| 8 | Absolute Maximum Ratings | 28 |
| 9 | RF Power Consumption Specifications | 28 |
| 10 | Wi-Fi Radio Characteristics | 29 |
| 11 | Receiver Characteristics – Basic Data Rate | 29 |
| 12 | Transmitter Characteristics – Basic Data Rate | 30 |
| 13 | Receiver Characteristics – Enhanced Data Rate | 30 |
| 14 | Transmitter Characteristics – Enhanced Data Rate | 31 |
| 15 | Receiver Characteristics – BLE | 32 |
| 16 | Transmitter Characteristics – BLE | 32 |
| 17 | ESP32 Ordering Information | 34 |
| 18 | Notes on ESP32 Pin Lists | 36 |
| 19 | GPIO_Matrix | 38 |
| 20 | Ethernet_MAC | 43 |

# List of Figures


| | | |
|---|---|---|
| 1 | Function Block Diagram | 5 |
| 2 | ESP32 Pin Layout (for QFN 6*6) | 6 |
| 3 | ESP32 Pin Layout (for QFN 5*5) | 7 |
| 4 | Address Mapping Structure | 13 |
| 5 | QFN48 (6x6 mm) Package | 33 |
| 6 | QFN48 (5x5 mm) Package | 33 |
| 7 | ESP32 Part Number | 34 |

# 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.

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 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.

## 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, please 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
- +12 dBm transmitting power
- NZIF receiver with -97 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: 2.3V to 3.6V

### 1.3.2 Clocks and Timers

- Internal 8 MHz oscillator with calibration
- Internal RC oscillator with calibration
- External 2 MHz to 60 MHz crystal oscillator (40 MHz only for Wi-Fi/BT functionality)
- 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 Learning Resources.

## 1.4 Applications

- 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

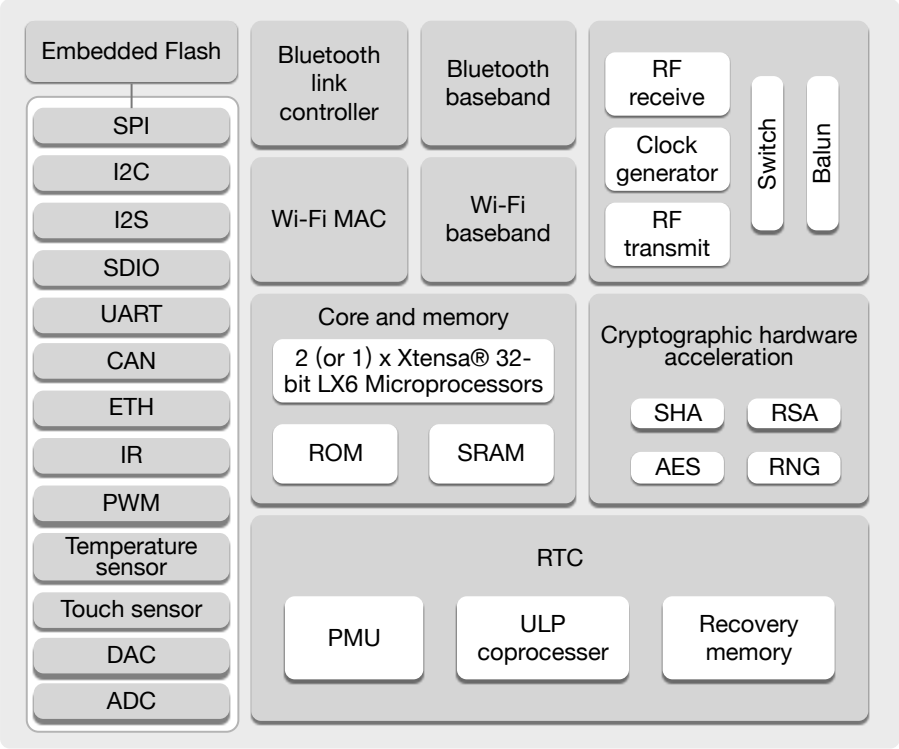



**Figure 1: Function Block Diagram**

**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

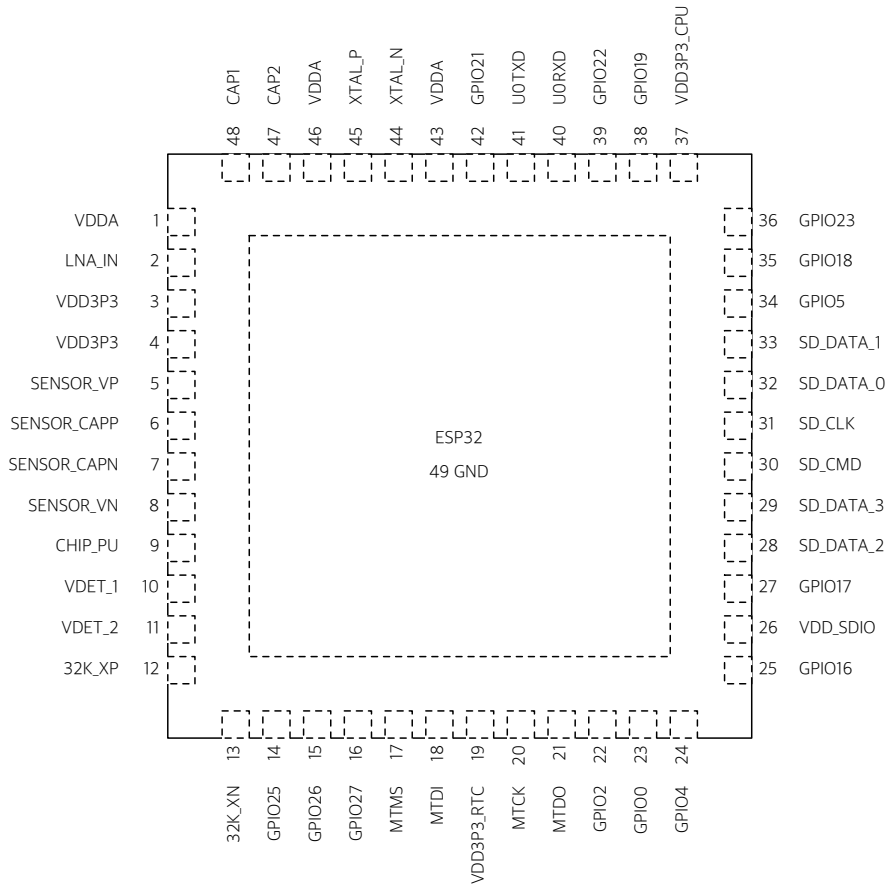



**Figure 2: ESP32 Pin Layout (for QFN 6*6)**

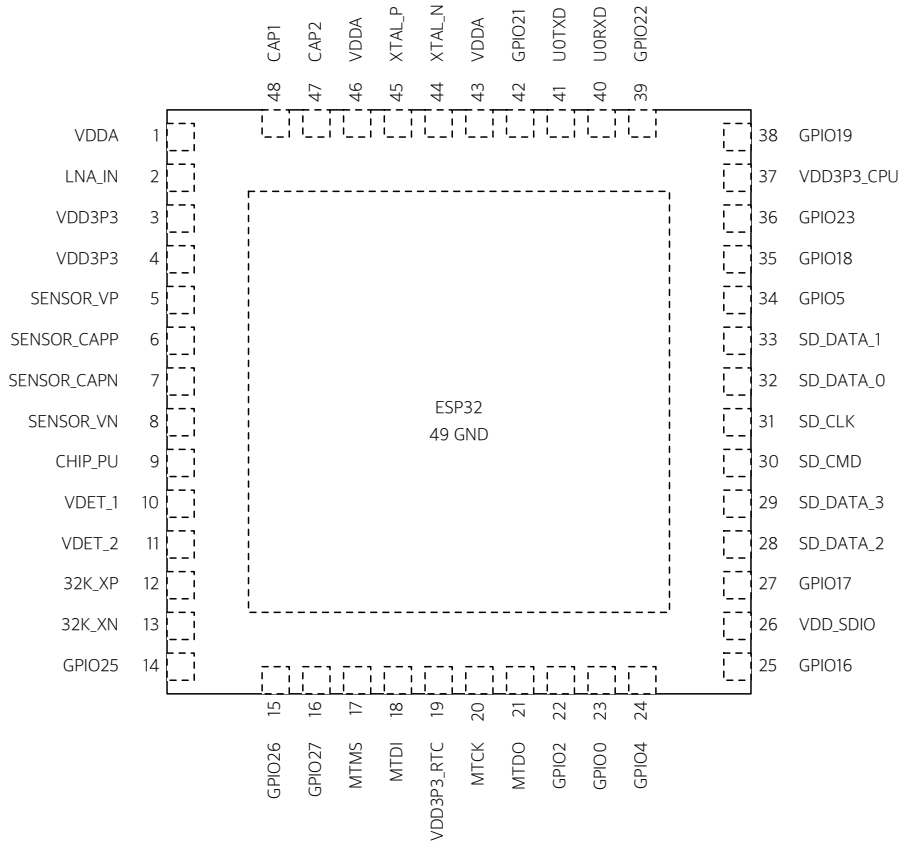



**Figure 3: ESP32 Pin Layout (for QFN 5*5)**

**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 2: 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 | Input power supply for RTC IO (1.8V ~ 3.6V) |
| 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 | Output power supply: 1.8V or the same voltage as VDD3P3_RTC |
| 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 |

| Name | No. | Type | Function |
|---|---|---|---|
| SD_CLK | 31 | I/O | GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS |
| 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 | Input power supply for CPU IO (1.8V ~ 3.6V) |
| 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 ESP32 Pin Lists.
- In most cases, the data port connection between the ESP32 and external flash is as follows: SD_DATA0/SPIQ = IO1/DO, SD_DATA1/SPID = IO0/DI, SD_DATA2/SPIHD = IO3/HOLD, SD_DATA3/SPIWP = IO2/WP.

## 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:**

- CHIP_PU must be activated after the 3.3V rails have been brought up. The recommended delay time (T) is given by the parameter of the RC circuit. For reference design, please refer to Figure **ESP-WROOM-32 Peripheral Schematics** in the *ESP-WROOM-32 Datasheet*.
- CHIP_PU is used to reset the chip. The input level to reset the chip should be below 0.6V and stays for at least 200 $\mu$s.
- 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
- GPIO0
- GPIO2
- MTDO
- GPIO5

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 3 for detailed boot modes configuration by strapping pins.

**Table 3: 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.
- The embedded flash operates at 1.8V. For the ESP32 series of chips that contain embedded flash, the MTDI should be pulled high.

# 3. Functional Description

This chapter describes the functions integrated in ESP32.

## 3.1 CPU and Memory

### 3.1.1 CPU

ESP32 contains one or 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 (160 MHz for ESP32-S0WD)
- 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 4.

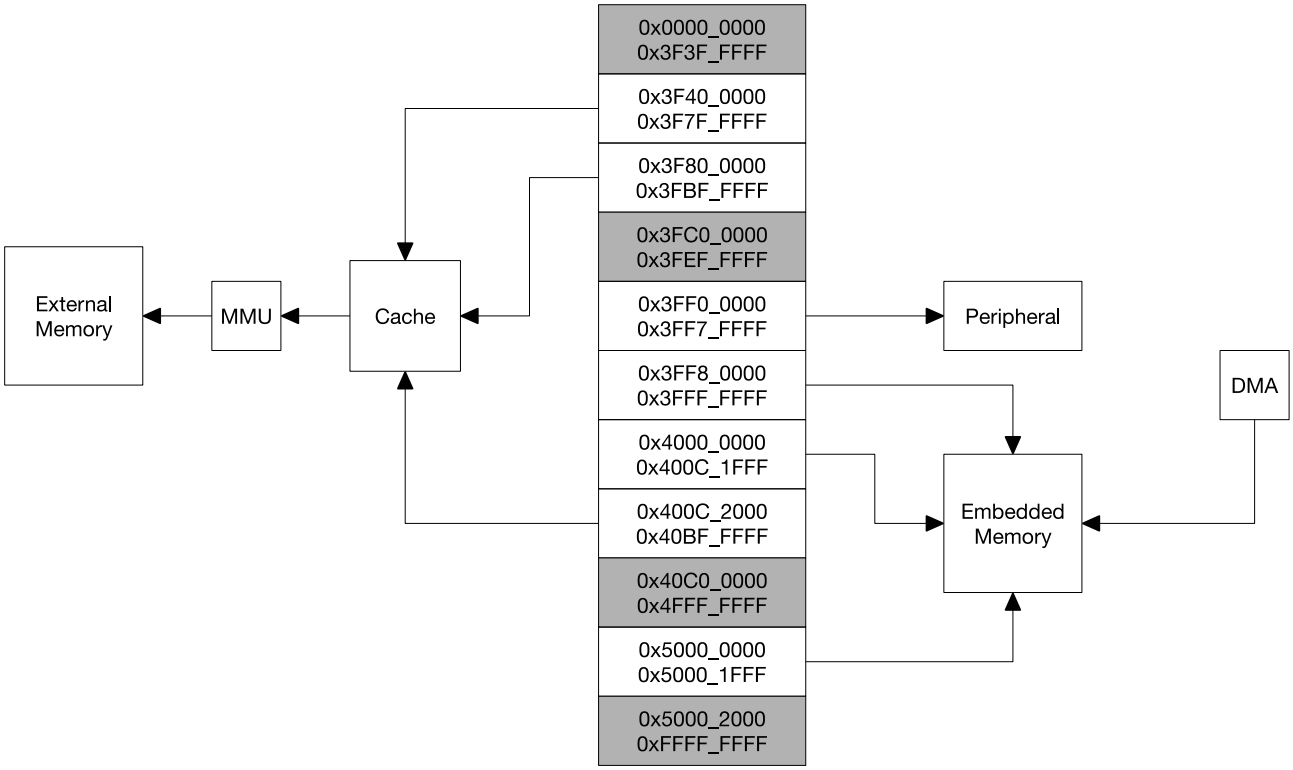



**Figure 4: Address Mapping Structure**

**Table 4: 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 three or four 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 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. More details can be found in the ESP32 Technical Reference Manual, in Chapter Reset and Clock.

## 3.4 Radio

The ESP32 radio consists of the following 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 two 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 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.

Passive or active scanning, as well as the P2P discovery procedure are performed autonomously when initiated by appropriate commands. Power management is handled with minimum host interaction to minimize active duty period.

### 3.5.1 Wi-Fi Radio and Baseband

The ESP32 Wi-Fi Radio and Baseband support the following features:

- 802.11b and 802.11g data rates
- 802.11n MCS0-7 in both 20 MHz and 40 MHz bandwidth
- 802.11n MCS32
- 802.11n 0.4 $\mu$s guard-interval
- up to 150 Mbps of data rate
- Receiving STBC 2x1
- Up to 20.5 dBm of transmitting power
- Adjustable transmitting power
- Antenna diversity and selection (software-managed hardware)

### 3.5.2 Wi-Fi MAC

The ESP32 Wi-Fi MAC applies low level protocol functions automatically, as follows:

- Request To Send (RTS), Clear To Send (CTS) and Acknowledgement (ACK/BA)
- Fragmentation and defragmentation
- Aggregation AMPDU and AMSDU
- WMM, U-APSD
- 802.11 e: QoS for wireless multimedia technology
- CCMP (CBC-MAC, counter mode), TKIP (MIC, RC4), WAPI (SMS4), WEP (RC4) and CRC
- Frame encapsulation (802.11h/RFC 1042)
- Automatic beacon monitoring/scanning

### 3.5.3 Wi-Fi Firmware

The ESP32 Wi-Fi Firmware provides the following functions:

- Infrastructure BSS Station mode / P2P mode / SoftAP mode support
- P2P Discovery, P2P Group Owner, P2P Group Client and P2P Power Management
- WPA/WPA2-Enterprise and WPS driver
- Additional 802.11i security features such as pre-authentication and TSN
- Open interface for various upper layer authentication schemes over EAP such as TLS, PEAP, LEAP, SIM, AKA or customer specific
- Clock/power gating combined with 802.11-compliant power management dynamically adapted to current connection condition providing minimal power consumption
- Adaptive rate fallback algorithm sets the optimal transmission rate and transmits power based on actual Signal Noise Ratio (SNR) and packet loss information
- Automatic retransmission and response on MAC to avoid packet discarding on slow host environment

### 3.5.4 Packet Traffic Arbitration (PTA)

ESP32 has a configurable Packet Traffic Arbitration (PTA) that provides flexible and exact timing Bluetooth co-existence support. It is a combination of both Frequency Division Multiplexing (FDM) and Time Division Multiplexing (TDM), and coordinates the protocol stacks.

## 3.6 Bluetooth

ESP32 integrates Bluetooth link controller and Bluetooth baseband, which carry out the baseband protocols and other low-level link routines, such as modulation/demodulation, packets processing, bit stream processing, frequency hopping, etc.

### 3.6.1 Bluetooth Radio and Baseband

The ESP32 Bluetooth Radio and Baseband support the following features:

- Class-1, class-2 and class-3 transmit output powers and up to 24 dB dynamic control range
- $\pi$/4 DQPSK and 8 DPSK modulation
- High performance in NZIF receiver sensitivity with over 97 dB dynamic range
- Class-1 operation without external PA
- Internal SRAM allows full speed data transfer, mixed voice and data, and full piconet operation
- Logic for forward error correction, header error control, access code correlation, CRC, demodulation, encryption bit stream generation, whitening and transmit pulse shaping
- ACL, SCO, eSCO and AFH
- A-law, $\mu$-law and CVSD digital audio CODEC in PCM interface
- SBC audio CODEC
- Power management for low power applications
- SMP with 128-bit AES

### 3.6.2 Bluetooth Interface

- Provides UART HCI interface, up to 4 Mbps
- Provides SDIO / SPI HCI interface
- Provides I2C interface for the host to do configuration
- Provides PCM / I2S audio interface

### 3.6.3 Bluetooth Stack

The Bluetooth stack of ESP32 is compliant with Bluetooth v4.2 BR / EDR and BLE specification.

### 3.6.4 Bluetooth Link Controller

The link controller operates in three major states: standby, connection and sniff. It enables multi connection and other operations like inquiry, page, and secure simple pairing, and therefore enables Piconet and Scatternet. Below are the features:

- Classic Bluetooth
  - Device Discovery (inquiry and inquiry scan)
  - Connection establishment (page and page scan)
  - Multi connections
  - Asynchronous data reception and transmission
  - Synchronous links (SCO/eSCO)
  - Master/Slave Switch
  - Adaptive Frequency Hopping and Channel assessment
  - Broadcast encryption
  - Authentication and encryption
  - Secure Simple Pairing
  - Multi-point and scatternet management
  - Sniff mode
  - Connectionless Slave Broadcast (transmitter and receiver)
  - Enhanced power control
  - Ping
- Bluetooth Low Energy
  - Advertising
  - Scanning
  - Multiple connections
  - Asynchronous data reception and transmission
  - Adaptive Frequency Hopping and Channel assessment
  - Connection parameter update
  - Date Length Extension
  - Link Layer Encryption
  - LE Ping

## 3.7 RTC and Low-Power Management

With the advanced power management technologies, ESP32 can switch between different power modes (see Table 5).

- Power mode
  - Active mode: The chip radio is powered on. The chip can receive, transmit, or listen.
  - Modem-sleep mode: The CPU is operational and the clock is configurable. The Wi-Fi/Bluetooth baseband and radio are disabled.

- Light-sleep mode: The CPU is paused. The RTC memory and RTC peripherals, as well as the ULP-coprocessor are running. Any wake-up events (MAC, host, RTC timer, or external interrupts) will wake up the chip.
- Deep-sleep mode: Only RTC memory and RTC peripherals are powered on. Wi-Fi and Bluetooth connection data are stored in RTC memory. The ULP-coprocessor can work.
- Hibernation mode: The internal 8-MHz oscillator and ULP-coprocessor are disabled. The RTC recovery memory is powered down. Only one RTC timer on the slow clock and some RTC GPIOs are active. The RTC timer or the RTC GPIOs can wake up the chip from the Hibernation mode.

• Sleep Pattern

- Association sleep pattern: The power mode switches between the Active mode, Modem- and Light-sleep mode during this sleep pattern. The CPU, Wi-Fi, Bluetooth, and radio are woken up at predetermined intervals to keep Wi-Fi/BT connections alive.
- ULP sensor-monitored pattern: The main CPU is in the Deep-sleep mode. The ULP co-processor does sensor measurements and wakes up the main system, based on the measured data from sensors.

**Table 5: Functionalities Depending on the Power Modes**

| Power mode | Active | Modem-sleep | Light-sleep | Deep-sleep | Hibernation |
|---|---|---|---|---|---|
| Sleep pattern | Association sleep pattern | | | ULP sensor-monitored pattern | - |
| CPU | ON | ON | PAUSE | OFF | OFF |
| Wi-Fi/BT baseband and radio | ON | OFF | OFF | OFF | OFF |
| RTC memory and RTC peripherals | ON | ON | ON | ON | OFF |
| ULP co-processor | ON | ON | ON | ON/OFF | OFF |

The power consumption varies with different power modes/sleep patterns, and work status, of functional modules. Please see Table 6 for details.

**Table 6: Power Consumption by Power Modes**

| Power mode | Description | Power consumption |
|---|---|---|
| Active (RF working) | Wi-Fi Tx packet 14 dBm ~ 19.5 dBm | Please refer to Table 9 for details. |
| | Wi-Fi / BT Tx packet 0 dBm | |
| | Wi-Fi / BT Rx and listening | |
| | Association sleep pattern (by Light-sleep) | 1 mA ~ 4 mA @DTIM3 |
| Modem-sleep | The CPU is powered on. | Max speed 240 MHz: 30 mA ~ 50 mA |
| | | Normal speed 80 MHz: 20 mA ~ 25 mA |
| | | Slow speed 2 MHz: 2 mA ~ 4 mA |
| Light-sleep | - | 0.8 mA |
| Deep-sleep | The ULP co-processor is powered on. | 150 $\mu$A |
| | ULP sensor-monitored pattern | 100 $\mu$A @1% duty |
| | RTC timer + RTC memory | 10 $\mu$A |
| Hibernation | RTC timer only | 5 $\mu$A |
| Power off | CHIP_PU is set to low level, the chip is powered off | 0.1 $\mu$A |

**Note:**

- During Deep-sleep, when ULP co-processor is powered on, peripherals such as GPIO and I2C are able to work.
- When the system works in the ULP sensor-monitored pattern, the ULP co-processor works with the ULP sensor periodically; ADC works with a duty cycle of 1%, so the power consumption is 100 $\mu$A.

# 4. Peripherals and Sensors

## 4.1 General Purpose Input / Output Interface (GPIO)

ESP32 has 34 GPIO pins which can be assigned to various functions by programming the appropriate registers. There are several kinds of GPIOs: digital only GPIOs, analog enabled GPIOs, capacitive touch enabled GPIOs, etc. Analog enabled GPIOs can be configured as digital GPIOs. Capacitive touch enabled GPIOs can be configured as digital GPIOs.

Most of the digital enabled GPIOs can be configured to internal pull-up or pull-down, or set to high impedance. When configured as an input, the input value can be read through the register. The input can also be set to edge-trigger or level-trigger to generate CPU interrupts. Most of the digital IO pins are bi-directional, non-inverting and tristate, including input and output buffer with tristate control. These pins can be multiplexed with other functions, such as the SDIO interface, UART, SPI, etc. (More details can be found in the Appendix, Table IO_MUX.) For low power operations, the GPIOs can be set to hold their states.

## 4.2 Analog-to-Digital Converter (ADC)

ESP32 integrates 12-bit SAR ADCs and supports measurements on 18 channels (analog enabled pins). Some of these pins can be used to build a programmable gain amplifier which is used for the measurement of small analog signals. The ULP-coprocessor in ESP32 is also designed to measure the voltages while operating in the sleep mode, to enable low power consumption; the CPU can be woken up by a threshold setting and/or via other triggers.

With the appropriate setting, the ADCs and the amplifier can be configured to measure voltages for a maximum of 18 pins.

## 4.3 Ultra-Low-Noise Analog Pre-Amplifier

ESP32 integrates an ultra-low-noise analog pre-amplifier that amplifies the voltage difference between pins SENSOR_VP and SENSOR_VN and outputs the value to the ADC. The amplification ratio is given by the size of a pair of sampling capacitors that are placed off-chip. By using a larger capacitor, the sampling noise is reduced, but the settling time will be increased. The amplification ratio is also limited by the amplifier which peaks at about 60 dB gain.

## 4.4 Hall Sensor

ESP32 integrates a Hall sensor based on an N-carrier resistor. When the chip is in the magnetic field, the Hall sensor develops a small voltage laterally on the resistor, which can be directly measured by the ADC, or amplified by the ultra-low-noise analog pre-amplifier and then measured by the ADC.

## 4.5 Digital-to-Analog Converter (DAC)

Two 8-bit DAC channels can be used to convert two digital signals into two analog voltage signal outputs. The design structure is composed of integrated resistor strings and a buffer. This dual DAC supports power supply as input voltage reference and can drive other circuits. The dual channels support independent conversions.

## 4.6 Temperature Sensor

The temperature sensor generates a voltage that varies with temperature. The voltage is internally converted via an analog-to-digital converter into a digital code.

The temperature sensor has a range of -40°C to 125°C. As the offset of the temperature sensor varies from chip to chip due to process variation, together with the heat generated by the Wi-Fi circuitry itself (which affects measurements), the internal temperature sensor is only suitable for applications that detect temperature changes instead of absolute temperatures and for calibration purposes as well.

However, if the user calibrates the temperature sensor and uses the device in a minimally powered-on application, the results could be accurate enough.

## 4.7 Touch Sensor

ESP32 offers 10 capacitive sensing GPIOs which detect capacitive variations introduced by the GPIO's direct contact or close proximity with a finger or other objects. The low noise nature of the design and high sensitivity of the circuit allow relatively small pads to be used. Arrays of pads can also be used so that a larger area or more points can be detected. The 10 capacitive sensing GPIOs are listed in Table 7.

**Table 7: Capacitive Sensing GPIOs Available on ESP32**

| Capacitive sensing signal name | Pin name |
|---|---|
| T0 | GPIO4 |
| T1 | GPIO0 |
| T2 | GPIO2 |
| T3 | MTDO |
| T4 | MTCK |
| T5 | MTD1 |
| T6 | MTMS |
| T7 | GPIO27 |
| T8 | 32K_XN |
| T9 | 32K_XP |

## 4.8 Ultra-Lower-Power Coprocessor

The ULP processor and RTC memory remains powered on during the Deep-sleep mode. Hence, the developer can store a program for the ULP processor in the RTC memory to access the peripheral devices, internal timers and internal sensors during the Deep-sleep mode. This is useful for designing applications where the CPU needs to be woken up by an external event, or timer, or a combination of these events, while maintaining minimal power consumption.

## 4.9 Ethernet MAC Interface

An IEEE-802.3-2008-compliant Media Access Controller (MAC) is provided for Ethernet LAN communications. ESP32 requires an external physical interface device (PHY) to connect to the physical LAN bus (twisted-pair, fiber,

etc.). The PHY is connected to ESP32 through 17 signals of MII or nine signals of RMII. With the Ethernet MAC (EMAC) interface, the following features are supported:

- 10 Mbps and 100 Mbps rates
- Dedicated DMA controller allowing high-speed transfer between the dedicated SRAM and Ethernet MAC
- Tagged MAC frame (VLAN support)
- Half-duplex (CSMA/CD) and full-duplex operation
- MAC control sublayer (control frames)
- 32-bit CRC generation and removal
- Several address filtering modes for physical and multicast address (multicast and group addresses)
- 32-bit status code for each transmitted or received frame
- Internal FIFOs to buffer transmit and receive frames. The transmit FIFO and the receive FIFO are both 512 words (32-bit)
- Hardware PTP (precision time protocol) in accordance with IEEE 1588 2008 (PTP V2)
- 25 MHz/50 MHz clock output

## 4.10 SD/SDIO/MMC Host Controller

An SD/SDIO/MMC host controller is available on ESP32 which supports the following features:

- Secure Digital memory (SD mem Version 3.0 and Version 3.01)
- Secure Digital I/O (SDIO Version 3.0)
- Consumer Electronics Advanced Transport Architecture (CE-ATA Version 1.1)
- Multimedia Cards (MMC Version 4.41, eMMC Version 4.5 and Version 4.51)

The controller allows clock output at up to 80 MHz and in three different data-bus modes: 1-bit, 4-bit and 8-bit. It supports two SD/SDIO/MMC4.41 cards in 4-bit data-bus mode. It also supports one SD card operating at 1.8 V level.

## 4.11 SDIO/SPI Slave Controller

ESP32 integrates an SD device interface that conforms to the industry-standard SDIO Card Specification Version 2.0 and allows a host controller to access the SoC device using the SDIO bus interface and protocol. ESP32 acts as the slave on the SDIO bus. The host can access SDIO interface registers directly and can access shared memory via a DMA engine, thus maximizing performance without engaging the processor cores.

The SDIO/SPI slave controller supports the following features:

- SPI, 1-bit SDIO, and 4-bit SDIO transfer modes over the full clock range of 0 to 50 MHz
- Configurable sampling and driving clock edge
- Special registers for direct access by host
- Interrupt to host for initiating data transfer
- Automatic loading of SDIO bus data and automatic discarding of padding data
- Block size of up to 512 bytes

- Interrupt vectors between the host and the slave to allow both to interrupt each other
- Supports DMA for data transfer

## 4.12 Universal Asynchronous Receiver Transmitter (UART)

ESP32 has three UART interfaces, i.e. UART0, UART1 and UART2, which provide asynchronous communication (RS232 and RS485) and IrDA support, and communicate at up to 5 Mbps. UART provides hardware management of the CTS and RTS signals and software flow control (XON and XOFF). All of the interfaces can be accessed by the DMA controller or directly by CPU.

## 4.13 I2C Interface

ESP32 has two I2C bus interfaces which can serve as I2C master or slave depending on the user's configuration. The I2C interfaces support:

- Standard mode (100 kbit/s)
- Fast mode (400 kbit/s)
- Up to 5 MHz, but constrained by SDA pull up strength
- 7-bit/10-bit addressing mode
- Dual addressing mode

Users can program command registers to control I2C interfaces to have more flexibility.

## 4.14 I2S Interface

Two standard I2S interfaces are available in ESP32. They can be operated in the master or slave mode, in full duplex and half-duplex communication modes, and can be configured to operate with an 8-/16-/32-/40-/48-bit resolution as input or output channels. BCK clock frequency from 10 kHz up to 40 MHz are supported. When one or both of the I2S interfaces are configured in the master mode, the master clock can be output to the external DAC/CODEC.

Both of the I2S interfaces have dedicated DMA controllers. PDM and BT PCM interfaces are supported.

## 4.15 Infrared Remote Controller

The infrared remote controller supports eight channels of infrared remote transmission and receiving. Through programming the pulse waveform, it supports various infrared protocols. Eight channels share a 512 x 32-bit block of memory to store the transmitting or receiving waveform.

## 4.16 Pulse Counter

The pulse counter captures pulse and counts pulse edges through seven modes. It has eight channels; each channel captures four signals at a time. The four input signals include two pulse signals and two control signals. When the counter reaches a defined threshold, an interrupt is generated.

## 4.17 Pulse Width Modulation (PWM)

The Pulse Width Modulation (PWM) controller can be used for driving digital motors and smart lights. The controller consists of PWM timers, the PWM operator and a dedicated capture sub-module. Each timer provides timing in synchronous or independent form, and each PWM operator generates the waveform for one PWM channel. The dedicated capture sub-module can accurately capture external timing events.

## 4.18 LED PWM

The LED PWM controller can generate 16 independent channels of digital waveforms with the configurable periods and configurable duties.

The 16 channels of digital waveforms operate at 80 MHz APB clock, among which 8 channels have the option of using the 8 MHz oscillator clock. Each channel can select a 20-bit timer with configurable counting range and its accuracy of duty can be up to 16 bits with the 1 ms period.

The software can change the duty immediately. Moreover, each channel supports step-by-step duty increasing or decreasing automatically. It is useful for the LED RGB color gradient generator.

## 4.19 Serial Peripheral Interface (SPI)

ESP32 features three SPIs (SPI, HSPI and VSPI) in slave and master modes in 1-line full-duplex and 1/2/4-line half-duplex communication modes. These SPIs also support the following general-purpose SPI features:

- 4 timing modes of the SPI format transfer that depend on the polarity (POL) and the phase (PHA)
- up to 80 MHz and the divided clocks of 80 MHz
- up to 64-byte FIFO

All SPIs can also be used to connect to the external flash/SRAM and LCD. Each SPI can be served by DMA controllers.

## 4.20 Accelerator

ESP32 is equipped with hardware accelerators of general algorithms, such as AES (FIPS PUB 197), SHA (FIPS PUB 180-4), RSA, and ECC, which support independent arithmetic such as Big Integer Multiplication and Big Integer Modular Multiplication. The maximum operation length for RSA, ECC, Big Integer Multiply and Big Integer Modular Multiplication is 4096 bits.

The hardware accelerators greatly improve operation speed and reduce software complexity. They also support code encryption and dynamic decryption which ensures that codes in the flash will not be stolen.

# 5. Electrical Characteristics

**Note:**
The specifications in this chapter have been tested under the following general condition: VDD = 3.3V, $T_A$ = 27°C, unless otherwise specified.

## 5.1 Absolute Maximum Ratings

**Table 8: Absolute Maximum Ratings**

| Parameter | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Power supply[1] | VDD | 2.3 | 3.3 | 3.6 | V |
| Minimum current delivered by power supply | $I_{VDD}$ | 0.5 | - | - | A |
| Input low voltage | $V_{IL}$ | -0.3 | - | 0.25×$V_{IO}$[2] | V |
| Input high voltage | $V_{IH}$ | 0.75×$V_{IO}$[2] | - | $V_{IO}$[2]+0.3 | V |
| Input leakage current | $I_{IL}$ | - | - | 50 | nA |
| Input pin capacitance | $C_{pad}$ | - | - | 2 | pF |
| Output low voltage | $V_{OL}$ | - | - | 0.1×$V_{IO}$[2] | V |
| Output high voltage | $V_{OH}$ | 0.8×$V_{IO}$[2] | - | - | V |
| Maximum output drive capability | $I_{MAX}$ | - | - | 40 | mA |
| Storage temperature range | $T_{STR}$ | -40 | - | 150 | °C |
| Operating temperature range[3] | $T_{OPR}$ | -40 | - | 125 | °C |

1. The power supplies include VDDA, VDD3P3, VDD3P3_RTC, VDD3P3_CPU, VDD_SDIO. The VDD_SDIO also supports 1.8V mode.
2. $V_{IO}$ is the power supply for a specific pad. More details can be found in Appendix, Table IO_MUX. For example, the power supply for SD_CLK is the VDD_SDIO.
3. Since the range of operating temperatures for the embedded flash on ESP32-D2WD is -40°C ~ 105°C, the operating temperatures for ESP32-D2WD extend from -40°C to 105°C. The other chips in this series have no embedded flash, and their range of operating temperatures is -40°C ~ 125°C.

## 5.2 RF Power Consumption Specifications

The power consumption measurements are conducted with 3.0V supply and 25°C ambient, at antenna port. All the transmitters' measurements are based on 50% duty cycle.

**Table 9: RF Power Consumption Specifications**

| Mode | Min | Typ | Max | Unit |
|---|---|---|---|---|
| Transmit 802.11b, DSSS 1 Mbps, POUT = +19.5 dBm | - | 240 | - | mA |
| Transmit 802.11b, OFDM 54 Mbps, POUT = +16 dBm | - | 190 | - | mA |
| Transmit 802.11g, OFDM MCS7, POUT = +14 dBm | - | 180 | - | mA |
| Receive 802.11b/g/n | - | 95 ~ 100 | - | mA |
| Transmit BT/BLE, POUT = 0 dBm | - | 130 | - | mA |
| Receive BT/BLE | - | 95 ~ 100 | - | mA |

## 5.3 Wi-Fi Radio

**Table 10: Wi-Fi Radio Characteristics**

| Description | Min | Typical | Max | Unit |
|---|---|---|---|---|
| Input frequency | 2412 | - | 2484 | MHz |
| Output impedance* | - | * | - | Ω |
| Input reflection | - | - | -10 | dB |
| Tx power | | | | |
| Output power of PA for 72.2 Mbps | 13 | 14 | 15 | dBm |
| Output power of PA for 11b mode | 19.5 | 20 | 20.5 | dBm |
| Sensitivity | | | | |
| DSSS, 1 Mbps | - | -98 | - | dBm |
| CCK, 11 Mbps | - | -91 | - | dBm |
| OFDM, 6 Mbps | - | -93 | - | dBm |
| OFDM, 54 Mbps | - | -75 | - | dBm |
| HT20, MCS0 | - | -93 | - | dBm |
| HT20, MCS7 | - | -73 | - | dBm |
| HT40, MCS0 | - | -90 | - | dBm |
| HT40, MCS7 | - | -70 | - | dBm |
| MCS32 | - | -89 | - | dBm |
| Adjacent channel rejection | | | | |
| OFDM, 6 Mbps | - | 37 | - | dB |
| OFDM, 54 Mbps | - | 21 | - | dB |
| HT20, MCS0 | - | 37 | - | dB |
| HT20, MCS7 | - | 20 | - | dB |

*The typical value of ESP32's Wi-Fi radio output impedance is different among chips of different QFN packages. For ESP32 chips with QFN 6x6 package (ESP32-D0WDQ6), the value is 30+j10 Ω; for ESP32 chips with QFN 5x5 package (ESP32-D0WD, ESP32-D2WD, ESP32-S0WD), the value is 35+j10 Ω.

## 5.4 Bluetooth Radio

### 5.4.1 Receiver–Basic Data Rate

**Table 11: Receiver Characteristics – Basic Data Rate**

| Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Sensitivity @0.1% BER | - | - | -94 | - | dBm |
| Maximum received signal @0.1% BER | - | 0 | - | - | dBm |
| Co-channel C/I | - | - | +7 | - | dB |
| Adjacent channel selectivity C/I | F = F0 + 1 MHz | - | - | -6 | dB |
| | F = F0 - 1 MHz | - | - | -6 | dB |
| | F = F0 + 2 MHz | - | - | -25 | dB |
| | F = F0 - 2 MHz | - | - | -33 | dB |
| | F = F0 + 3 MHz | - | - | -25 | dB |
| | F = F0 - 3 MHz | - | - | -45 | dB |

| Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Out-of-band blocking performance | 30 MHz ~ 2000 MHz | -10 | - | - | dBm |
| | 2000 MHz ~ 2400 MHz | -27 | - | - | dBm |
| | 2500 MHz ~ 3000 MHz | -27 | - | - | dBm |
| | 3000 MHz ~ 12.5 GHz | -10 | - | - | dBm |
| Intermodulation | - | -36 | - | - | dBm |

### 5.4.2 Transmitter – Basic Data Rate

**Table 12: Transmitter Characteristics – Basic Data Rate**

| Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| RF transmit power | - | - | 0 | - | dBm |
| Gain control step | - | - | ±3 | - | dBm |
| RF power control range | - | -12 | - | +12 | dBm |
| +12 dB bandwidth | - | - | 0.9 | - | MHz |
| Adjacent channel transmit power | F = F0 + 1 MHz | - | -24 | - | dBm |
| | F = F0 - 1 MHz | - | -16.1 | - | dBm |
| | F = F0 + 2 MHz | - | -40.8 | - | dBm |
| | F = F0 - 2 MHz | - | -35.6 | - | dBm |
| | F = F0 + 3 MHz | - | -45.7 | - | dBm |
| | F = F0 - 3 MHz | - | -40.2 | - | dBm |
| | F = F0 + > 3 MHz | - | -45.6 | - | dBm |
| | F = F0 - > 3 MHz | - | -44.6 | - | dBm |
| $\Delta f1_{avg}$ | - | - | - | 155 | kHz |
| $\Delta f2_{max}$ | - | 133.7 | - | - | kHz |
| $\Delta f2_{avg}/\Delta f1_{avg}$ | - | - | 0.92 | - | - |
| ICFT | - | - | -7 | - | kHz |
| Drift rate | - | - | 0.7 | - | kHz/50 $\mu$s |
| Drift (1 slot packet) | - | - | 6 | - | kHz |
| Drift (5 slot packet) | - | - | 6 | - | kHz |

### 5.4.3 Receiver–Enhanced Data Rate

**Table 13: Receiver Characteristics – Enhanced Data Rate**

| Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $\pi$/4 DQPSK | | | | | |
| Sensitivity @0.01% BER | - | - | -90 | - | dBm |
| Maximum received signal @0.01% BER | - | - | 0 | - | dBm |
| Co-channel C/I | - | - | 11 | - | dB |
| Adjacent channel selectivity C/I | F = F0 + 1 MHz | - | -7 | - | dB |
| | F = F0 - 1 MHz | - | -7 | - | dB |
| | F = F0 + 2 MHz | - | -25 | - | dB |
| | F = F0 - 2 MHz | - | -35 | - | dB |
| | F = F0 + 3 MHz | - | -25 | - | dB |
| | F = F0 - 3 MHz | - | -45 | - | dB |

| Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| 8DPSK | | | | | |
| Sensitivity @0.01% BER | - | - | -84 | - | dBm |
| Maximum received signal @0.01% BER | - | - | -5 | - | dBm |
| C/I c-channel | - | - | 18 | - | dB |
| Adjacent channel selectivity C/I | F = F0 + 1 MHz | - | 2 | - | dB |
| | F = F0 - 1 MHz | - | 2 | - | dB |
| | F = F0 + 2 MHz | - | -25 | - | dB |
| | F = F0 - 2 MHz | - | -25 | - | dB |
| | F = F0 + 3 MHz | - | -25 | - | dB |
| | F = F0 - 3 MHz | - | -38 | - | dB |

### 5.4.4 Transmitter–Enhanced Data Rate

**Table 14: Transmitter Characteristics – Enhanced Data Rate**

| Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| RF transmit power | - | - | 0 | - | dBm |
| Gain control step | - | - | ±3 | - | dBm |
| RF power control range | - | -12 | - | +12 | dBm |
| $\pi$/4 DQPSK max w0 | - | - | -0.72 | - | kHz |
| $\pi$/4 DQPSK max wi | - | - | -6 | - | kHz |
| $\pi$/4 DQPSK max \|wi + w0\| | - | - | -7.42 | - | kHz |
| 8DPSK max w0 | - | - | 0.7 | - | kHz |
| 8DPSK max wi | - | - | -9.6 | - | kHz |
| 8DPSK max \|wi + w0\| | - | - | -10 | - | kHz |
| $\pi$/4 DQPSK modulation accuracy | RMS DEVM | - | 4.28 | - | % |
| | 99% DEVM | - | - | 30 | % |
| | Peak DEVM | - | 13.3 | - | % |
| 8 DPSK modulation accuracy | RMS DEVM | - | 5.8 | - | % |
| | 99% DEVM | - | - | 20 | % |
| | Peak DEVM | - | 14 | - | % |
| In-band spurious emissions | F = F0 + 1 MHz | - | -34 | - | dBm |
| | F = F0 - 1 MHz | - | -40.2 | - | dBm |
| | F = F0 + 2 MHz | - | -34 | - | dBm |
| | F = F0 - 2 MHz | - | -36 | - | dBm |
| | F = F0 + 3 MHz | - | -38 | - | dBm |
| | F = F0 - 3 MHz | - | -40.3 | - | dBm |
| | F = F0 +/- > 3 MHz | - | - | -41.5 | dBm |
| EDR differential phase coding | - | - | 100 | - | % |

## 5.5 Bluetooth LE Radio

### 5.5.1 Receiver

**Table 15: Receiver Characteristics – BLE**

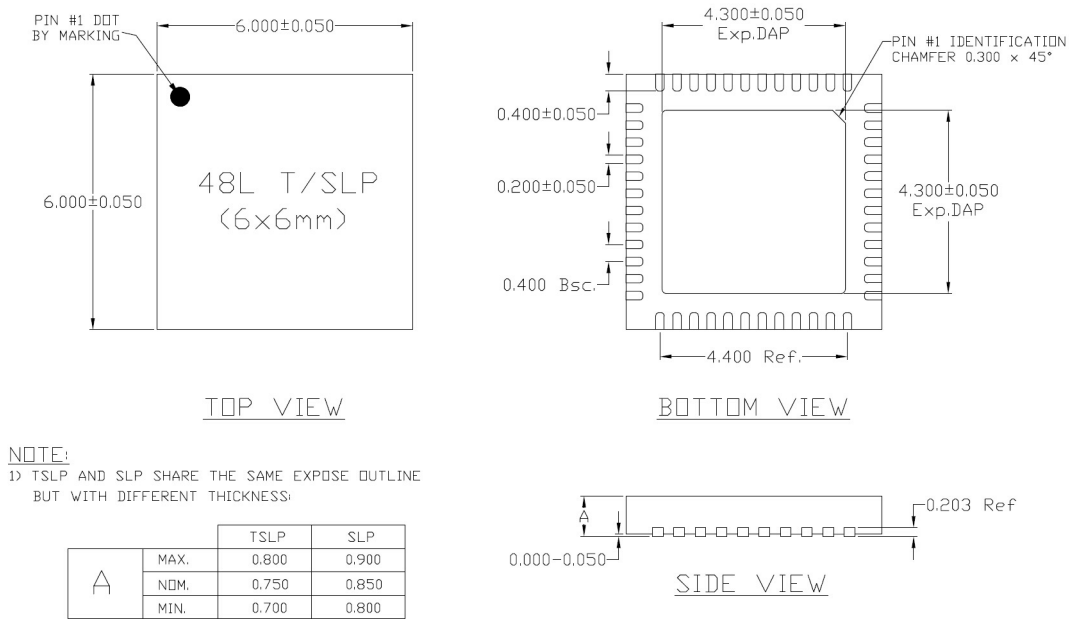
| Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Sensitivity @30.8% PER | - | - | -97 | - | dBm |
| Maximum received signal @30.8% PER | - | 0 | - | - | dBm |
| Co-channel C/I | - | - | +10 | - | dB |
| Adjacent channel selectivity C/I | F = F0 + 1 MHz | - | -5 | - | dB |
| | F = F0 - 1 MHz | - | -5 | - | dB |
| | F = F0 + 2 MHz | - | -25 | - | dB |
| | F = F0 - 2 MHz | - | -35 | - | dB |
| | F = F0 + 3 MHz | - | -25 | - | dB |
| | F = F0 - 3 MHz | - | -45 | - | dB |
| Out-of-band blocking performance | 30 MHz ~ 2000 MHz | -10 | - | - | dBm |
| | 2000 MHz ~ 2400 MHz | -27 | - | - | dBm |
| | 2500 MHz ~ 3000 MHz | -27 | - | - | dBm |
| | 3000 MHz ~ 12.5 GHz | -10 | - | - | dBm |
| Intermodulation | - | -36 | - | - | dBm |

### 5.5.2 Transmitter

**Table 16: Transmitter Characteristics – BLE**

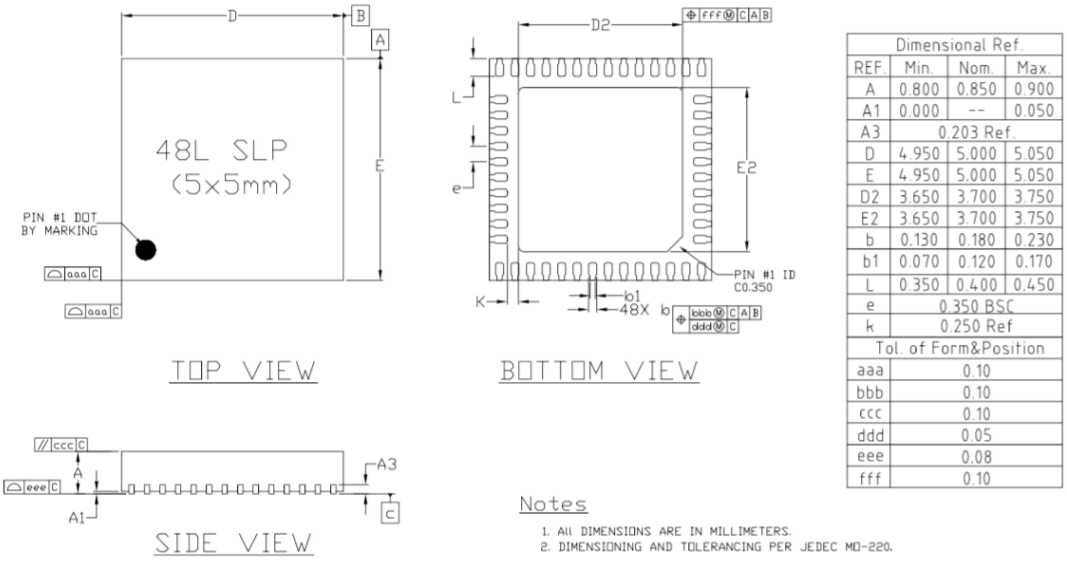
| Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| RF transmit power | - | - | 0 | - | dBm |
| Gain control step | - | - | ±3 | - | dBm |
| RF power control range | - | -12 | - | +12 | dBm |
| Adjacent channel transmit power | F = F0 + 1 MHz | - | -14.6 | - | dBm |
| | F = F0 - 1 MHz | - | -12.7 | - | dBm |
| | F = F0 + 2 MHz | - | -44.3 | - | dBm |
| | F = F0 - 2 MHz | - | -38.7 | - | dBm |
| | F = F0 + 3 MHz | - | -49.2 | - | dBm |
| | F = F0 - 3 MHz | - | -44.7 | - | dBm |
| | F = F0 + > 3 MHz | - | -50 | - | dBm |
| | F = F0 - > 3 MHz | - | -50 | - | dBm |
| $\Delta f1_{avg}$ | - | - | - | 265 | kHz |
| $\Delta f2_{max}$ | - | 247 | - | - | kHz |
| $\Delta f2_{avg}/\Delta f1_{avg}$ | - | - | -0.92 | - | - |
| ICFT | - | - | -10 | - | kHz |
| Drift rate | - | - | 0.7 | - | kHz/50 $\mu$s |
| Drift | - | - | 2 | - | kHz |

# 6. Package Information




| | | TSLP | SLP |
|---|---|---|---|
| A | MAX. | 0.800 | 0.900 |
| | NOM. | 0.750 | 0.850 |
| | MIN. | 0.700 | 0.800 |

**Figure 5: QFN48 (6x6 mm) Package**




| Dimensional Ref. | | | |
|---|---|---|---|
| REF. | Min. | Nom. | Max. |
| A | 0.800 | 0.850 | 0.900 |
| A1 | 0.000 | -- | 0.050 |
| A3 | 0.203 Ref. | | |
| D | 4.950 | 5.000 | 5.050 |
| E | 4.950 | 5.000 | 5.050 |
| D2 | 3.650 | 3.700 | 3.750 |
| E2 | 3.650 | 3.700 | 3.750 |
| b | 0.130 | 0.180 | 0.230 |
| b1 | 0.070 | 0.120 | 0.170 |
| L | 0.350 | 0.400 | 0.450 |
| e | 0.350 BSC | | |
| k | 0.250 Ref | | |
| Tol. of Form&Position | | | |
| aaa | 0.10 | | |
| bbb | 0.10 | | |
| ccc | 0.10 | | |
| ddd | 0.05 | | |
| eee | 0.08 | | |
| fff | 0.10 | | |

**Figure 6: QFN48 (5x5 mm) Package**

# 7. Part Number and Ordering Information

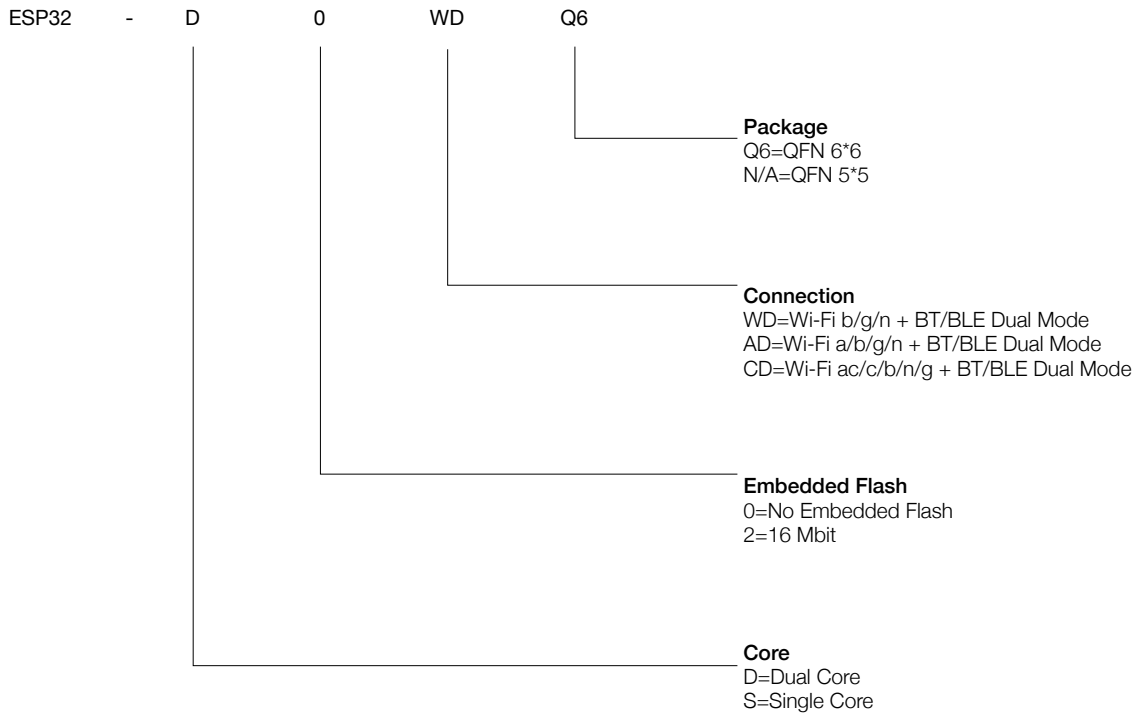



**Figure 7: ESP32 Part Number**

The table below provides the ordering information of the ESP32 series of chips.

**Table 17: ESP32 Ordering Information**

| Ordering code | Core | Embedded flash | Connection | Package |
|---|---|---|---|---|
| ESP32-D0WDQ6 | Dual core | No embedded flash | Wi-Fi b/g/n + BT/BLE Dual Mode | QFN 6*6 |
| ESP32-D0WD | Dual core | No embedded flash | Wi-Fi b/g/n + BT/BLE Dual Mode | QFN 5*5 |
| ESP32-D2WD | Dual core | 16-Mbit embedded flash | Wi-Fi b/g/n + BT/BLE Dual Mode | QFN 5*5 |
| ESP32-S0WD | Single core | No embedded flash | Wi-Fi b/g/n + BT/BLE Dual Mode | QFN 5*5 |

# 8. Learning Resources

## 8.1 Must-Read Documents

Click on the following links for related documents of ESP32.

- ESP32 Technical Reference Manual
  The manual provides detailed information on how to use the ESP32 memory and peripherals.
- ESP32 Hardware Resources
  The zip files include the schematics, PCB layout, Gerber and BOM list of ESP32-DevKitC.
- ESP32 Hardware Design Guidelines
  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.
- ESP32 AT Instruction Set and Examples
  This document introduces the ESP32 AT commands, explains how to use them and provides examples of several common AT commands.

## 8.2 Must-Have Resources

Here are the ESP32-related must-have resources.

- ESP32 BBS
  This is an Engineer-to-Engineer (E2E) Community for ESP32 where you can post questions, share knowledge, explore ideas, and help solve problems with fellow engineers.
- ESP32 Github
  ESP32 development projects are freely distributed under Espressif's MIT license on Github. It is established to help developers get started with ESP32 and foster innovation and the growth of general knowledge about the hardware and software surrounding ESP32 devices.
- ESP32 Tools
  This is a web-page where users can download ESP32 Flash Download Tools and the zip file "ESP32 Certification and Test".
- ESP32 IDF
  This web-page links users to the official IoT development framework for ESP32.
- ESP32 Resources
  This webpage provides the links to all the available ESP32 documents, SDK and tools.

# Appendix A – ESP32 Pin Lists

## A.1. Notes on ESP32 Pin Lists

**Table 18: Notes on ESP32 Pin Lists**

| No. | Description |
|---|---|
| 1 | In Table IO_MUX, the red-filled areas mark the differences from ESP31B. The blue-filled areas indicate the new features of ESP32, compared to those of ESP31B. The yellow-filled areas indicate the GPIO pins that are input-only. Please see the next note for details. |
| 2 | GPIO pins 34-39 are input-only. These pins do not feature an output driver or internal pull-up/pull-down circuitry. The pin names are: SENSOR_VP (GPIO36), SENSOR_CAPP (GPIO37), SENSOR_CAPN (GPIO38), SENSOR_VN (GPIO39), VDET_1 (GPIO34), VDET_2 (GPIO35). |
| 3 | The pins are split into four power domains: VANA (analog power supply), VRTC (RTC power supply), VIO (power supply of digital IOs and CPU cores), VSDIO (power supply of SDIO IOs). VSDIO is the output of the internal SDIO-LDO. The voltage of SDIO-LDO can be configured at 1.8V, or be the same as that of the VRTC. The strapping pin and eFuse bits determine the default voltage of the SDIO-LDO. Software can change the voltage of the SDIO-LDO by configuring register bits. For details, please see the column "Power Domain" in Table IO_MUX. |
| 4 | The functional pins in the VRTC domain are those with analog functions, including the 32 kHz crystal oscillator, ADC pre-amplifier, ADC, DAC, and capacitive touch sensor. Please see columns "Analog Function 1~3" in Table IO_MUX. |
| 5 | These VRTC pins support the RTC function, and can work during Deep-sleep. For example, an RTC-GPIO can be used for waking up the chip from Deep-sleep. |
| 6 | The GPIO pins support up to six digital functions, as shown in columns "Function 1~6" In Table IO_MUX. The function selection registers will be set as "*N*-1", where *N* is the function number. Below are some definitions:<br>• SD_* is for signals of the SDIO slave.<br>• HS1_* is for Port 1 signals of SDIO host.<br>• HS2_* is for Port 2 signals of SDIO host.<br>• MT* is for signals of the JTAG.<br>• U0* is for signals of the UART0 module.<br>• U1* is for signals of the UART1 module.<br>• U2* is for signals of the UART2 module.<br>• SPI* is for signals of the SPI01 module.<br>• HSPI* is for signals of the SPI2 module.<br>• VSPI* is for signals of the SPI3 module. |

| No. | Description |
|---|---|
| 7 | Each digital "Function" column is accompanied by a column of "Type". Please see the following explanations for the meaning of "type" with respect to each "function" it is associated with. For any "Function-*N*", "type" signifies:<br>• I: input only. If a function other than "Function-*N*" is assigned, the input signal of "Function-*N*" is still from this pin.<br>• I1: input only. If a function other than "Function-*N*" is assigned, the input signal for "Function-*N*" is always "1".<br>• I0: input only. If a function other than "Function-*N*" is assigned, the input signal for "Function-*N*" is always "0".<br>• O: output only.<br>• T: high-impedance.<br>• I/O/T: combinations of input, output, and high-impedance according to the function signal.<br>• I1/O/T: combinations of input, output, and high-impedance according to the function signal. If a function is not selected, the input signal of the function is "1".<br>For example, pin 30 can act as HS1_CMD or SD_CMD, where HS1_CMD is of an "I1/O/T" type. If pin 30 is selected as HS1_CMD, the input and output of this pin are controlled by the SDIO host. If pin 30 is not selected as HS1_CMD, the input signal to SDIO host is always "1". |
| 8 | Each digital output pin is associated with its configurable drive-strength. Column "Drive Strength" in Table IO_MUX lists the default values. The drive strength of the digital output pins can be configured into one of the following four options:<br>• 0: ~5 mA<br>• 1: ~10 mA<br>• 2: ~20 mA<br>• 3: ~40 mA<br>The default value is 2.<br>The drive strength of the internal pull-up (wpu) and pull-down (wpd) is ~75 $\mu$A. |
| 9 | Column "At Reset" in Table IO_MUX lists the status of each pin during reset, including input enable (ie=1), internal pull-up (wpu) and internal pull-down (wpd). During reset, all pins are output-disabled. |
| 10 | Column "After Reset" in Table IO_MUX lists the status of each pin immediately after reset, including input enable (ie=1), internal pull-up (wpu) and internal pull-down (wpd). After reset, each pin is set to its "Function 1". The output enable are controlled by its digital Function 1. |
| 11 | Table Ethernet_MAC is about the signal mapping inside Ethernet MAC. The Ethernet MAC supports MII and RMII interfaces, and supports both internal PLL clock and the external clock source. For MII interface, the Ethernet MAC is with/without the TX_ERR signal. MDC, MDIO, CRS and COL are slow signals, and can be mapped onto any GPIO pins through GPIO-Matrix. |
| 12 | Table GPIO Matrix is for the GPIO-Matrix. The signals of the on-chip functional modules can be mapped onto any GPIO pins. Some signals can be mapped onto a pin by both IO-MUX and GPIO-Matrix, as shown in the column tagged as "Same input signal from IO_MUX core" in Table GPIO Matrix. |
| 13 | *In Table GPIO_Matrix the column "Default Value if unassigned" records the default value of the an input signal if no GPIO is assigned to it. The actual value is determined by register GPIO_FUNC*m*_IN_INV_SEL and GPIO_FUNC*m*_IN_SEL. (The value of *m* ranges from 1 to 255.) |

## A.2. GPIO_Matrix

**Table 19: GPIO_Matrix**

| Signal No. | Input signals | Default value if unassigned* | Same input signal from IO_MUX core | Output signals | Output enable of output signals |
|---|---|---|---|---|---|
| 0 | SPICLK_in | 0 | yes | SPICLK_out | SPICLK_oe |
| 1 | SPIQ_in | 0 | yes | SPIQ_out | SPIQ_oe |
| 2 | SPID_in | 0 | yes | SPID_out | SPID_oe |
| 3 | SPIHD_in | 0 | yes | SPIHD_out | SPIHD_oe |
| 4 | SPIWP_in | 0 | yes | SPIWP_out | SPIWP_oe |
| 5 | SPICS0_in | 0 | yes | SPICS0_out | SPICS0_oe |
| 6 | SPICS1_in | 0 | no | SPICS1_out | SPICS1_oe |
| 7 | SPICS2_in | 0 | no | SPICS2_out | SPICS2_oe |
| 8 | HSPICLK_in | 0 | yes | HSPICLK_out | HSPICLK_oe |
| 9 | HSPIQ_in | 0 | yes | HSPIQ_out | HSPIQ_oe |
| 10 | HSPID_in | 0 | yes | HSPID_out | HSPID_oe |
| 11 | HSPICS0_in | 0 | yes | HSPICS0_out | HSPICS0_oe |
| 12 | HSPIHD_in | 0 | yes | HSPIHD_out | HSPIHD_oe |
| 13 | HSPIWP_in | 0 | yes | HSPIWP_out | HSPIWP_oe |
| 14 | U0RXD_in | 0 | yes | U0TXD_out | 1'd1 |
| 15 | U0CTS_in | 0 | yes | U0RTS_out | 1'd1 |
| 16 | U0DSR_in | 0 | no | U0DTR_out | 1'd1 |
| 17 | U1RXD_in | 0 | yes | U1TXD_out | 1'd1 |
| 18 | U1CTS_in | 0 | yes | U1RTS_out | 1'd1 |
| 23 | I2S0O_BCK_in | 0 | no | I2S0O_BCK_out | 1'd1 |
| 24 | I2S1O_BCK_in | 0 | no | I2S1O_BCK_out | 1'd1 |
| 25 | I2S0O_WS_in | 0 | no | I2S0O_WS_out | 1'd1 |
| 26 | I2S1O_WS_in | 0 | no | I2S1O_WS_out | 1'd1 |
| 27 | I2S0I_BCK_in | 0 | no | I2S0I_BCK_out | 1'd1 |
| 28 | I2S0I_WS_in | 0 | no | I2S0I_WS_out | 1'd1 |
| 29 | I2CEXT0_SCL_in | 1 | no | I2CEXT0_SCL_out | 1'd1 |
| 30 | I2CEXT0_SDA_in | 1 | no | I2CEXT0_SDA_out | 1'd1 |
| 31 | pwm0_sync0_in | 0 | no | sdio_tohost_int_out | 1'd1 |
| 32 | pwm0_sync1_in | 0 | no | pwm0_out0a | 1'd1 |
| 33 | pwm0_sync2_in | 0 | no | pwm0_out0b | 1'd1 |
| 34 | pwm0_f0_in | 0 | no | pwm0_out1a | 1'd1 |
| 35 | pwm0_f1_in | 0 | no | pwm0_out1b | 1'd1 |
| 36 | pwm0_f2_in | 0 | no | pwm0_out2a | 1'd1 |
| 37 | - | 0 | no | pwm0_out2b | 1'd1 |
| 39 | pcnt_sig_ch0_in0 | 0 | no | - | 1'd1 |
| 40 | pcnt_sig_ch1_in0 | 0 | no | - | 1'd1 |
| 41 | pcnt_ctrl_ch0_in0 | 0 | no | - | 1'd1 |
| 42 | pcnt_ctrl_ch1_in0 | 0 | no | - | 1'd1 |

| Signal No. | Input signals | Default value if unassigned | Same input signal from IO_MUX core | Output signals | Output enable of output signals |
|---|---|---|---|---|---|
| 43 | pcnt_sig_ch0_in1 | 0 | no | - | 1'd1 |
| 44 | pcnt_sig_ch1_in1 | 0 | no | - | 1'd1 |
| 45 | pcnt_ctrl_ch0_in1 | 0 | no | - | 1'd1 |
| 46 | pcnt_ctrl_ch1_in1 | 0 | no | - | 1'd1 |
| 47 | pcnt_sig_ch0_in2 | 0 | no | - | 1'd1 |
| 48 | pcnt_sig_ch1_in2 | 0 | no | - | 1'd1 |
| 49 | pcnt_ctrl_ch0_in2 | 0 | no | - | 1'd1 |
| 50 | pcnt_ctrl_ch1_in2 | 0 | no | - | 1'd1 |
| 51 | pcnt_sig_ch0_in3 | 0 | no | - | 1'd1 |
| 52 | pcnt_sig_ch1_in3 | 0 | no | - | 1'd1 |
| 53 | pcnt_ctrl_ch0_in3 | 0 | no | - | 1'd1 |
| 54 | pcnt_ctrl_ch1_in3 | 0 | no | - | 1'd1 |
| 55 | pcnt_sig_ch0_in4 | 0 | no | - | 1'd1 |
| 56 | pcnt_sig_ch1_in4 | 0 | no | - | 1'd1 |
| 57 | pcnt_ctrl_ch0_in4 | 0 | no | - | 1'd1 |
| 58 | pcnt_ctrl_ch1_in4 | 0 | no | - | 1'd1 |
| 61 | HSPICS1_in | 0 | no | HSPICS1_out | HSPICS1_oe |
| 62 | HSPICS2_in | 0 | no | HSPICS2_out | HSPICS2_oe |
| 63 | VSPICLK_in | 0 | yes | VSPICLK_out_mux | VSPICLK_oe |
| 64 | VSPIQ_in | 0 | yes | VSPIQ_out | VSPIQ_oe |
| 65 | VSPID_in | 0 | yes | VSPID_out | VSPID_oe |
| 66 | VSPIHD_in | 0 | yes | VSPIHD_out | VSPIHD_oe |
| 67 | VSPIWP_in | 0 | yes | VSPIWP_out | VSPIWP_oe |
| 68 | VSPICS0_in | 0 | yes | VSPICS0_out | VSPICS0_oe |
| 69 | VSPICS1_in | 0 | no | VSPICS1_out | VSPICS1_oe |
| 70 | VSPICS2_in | 0 | no | VSPICS2_out | VSPICS2_oe |
| 71 | pcnt_sig_ch0_in5 | 0 | no | ledc_hs_sig_out0 | 1'd1 |
| 72 | pcnt_sig_ch1_in5 | 0 | no | ledc_hs_sig_out1 | 1'd1 |
| 73 | pcnt_ctrl_ch0_in5 | 0 | no | ledc_hs_sig_out2 | 1'd1 |
| 74 | pcnt_ctrl_ch1_in5 | 0 | no | ledc_hs_sig_out3 | 1'd1 |
| 75 | pcnt_sig_ch0_in6 | 0 | no | ledc_hs_sig_out4 | 1'd1 |
| 76 | pcnt_sig_ch1_in6 | 0 | no | ledc_hs_sig_out5 | 1'd1 |
| 77 | pcnt_ctrl_ch0_in6 | 0 | no | ledc_hs_sig_out6 | 1'd1 |
| 78 | pcnt_ctrl_ch1_in6 | 0 | no | ledc_hs_sig_out7 | 1'd1 |
| 79 | pcnt_sig_ch0_in7 | 0 | no | ledc_ls_sig_out0 | 1'd1 |
| 80 | pcnt_sig_ch1_in7 | 0 | no | ledc_ls_sig_out1 | 1'd1 |
| 81 | pcnt_ctrl_ch0_in7 | 0 | no | ledc_ls_sig_out2 | 1'd1 |
| 82 | pcnt_ctrl_ch1_in7 | 0 | no | ledc_ls_sig_out3 | 1'd1 |
| 83 | rmt_sig_in0 | 0 | no | ledc_ls_sig_out4 | 1'd1 |
| 84 | rmt_sig_in1 | 0 | no | ledc_ls_sig_out5 | 1'd1 |
| 85 | rmt_sig_in2 | 0 | no | ledc_ls_sig_out6 | 1'd1 |

| Signal No. | Input signals | Default value if unassigned | Same input signal from IO_MUX core | Output signals | Output enable of output signals |
|---|---|---|---|---|---|
| 86 | rmt_sig_in3 | 0 | no | ledc_ls_sig_out7 | 1'd1 |
| 87 | rmt_sig_in4 | 0 | no | rmt_sig_out0 | 1'd1 |
| 88 | rmt_sig_in5 | 0 | no | rmt_sig_out1 | 1'd1 |
| 89 | rmt_sig_in6 | 0 | no | rmt_sig_out2 | 1'd1 |
| 90 | rmt_sig_in7 | 0 | no | rmt_sig_out3 | 1'd1 |
| 91 | - | - | - | rmt_sig_out4 | 1'd1 |
| 92 | - | - | - | rmt_sig_out6 | 1'd1 |
| 94 | - | - | - | rmt_sig_out7 | 1'd1 |
| 95 | I2CEXT1_SCL_in | 1 | no | I2CEXT1_SCL_out | 1'd1 |
| 96 | I2CEXT1_SDA_in | 1 | no | I2CEXT1_SDA_out | 1'd1 |
| 97 | host_card_detect_n_1 | 0 | no | host_ccmd_od_pullup_en_n | 1'd1 |
| 98 | host_card_detect_n_2 | 0 | no | host_rst_n_1 | 1'd1 |
| 99 | host_card_write_prt_1 | 0 | no | host_rst_n_2 | 1'd1 |
| 100 | host_card_write_prt_2 | 0 | no | gpio_sd0_out | 1'd1 |
| 101 | host_card_int_n_1 | 0 | no | gpio_sd1_out | 1'd1 |
| 102 | host_card_int_n_2 | 0 | no | gpio_sd2_out | 1'd1 |
| 103 | pwm1_sync0_in | 0 | no | gpio_sd3_out | 1'd1 |
| 104 | pwm1_sync1_in | 0 | no | gpio_sd4_out | 1'd1 |
| 105 | pwm1_sync2_in | 0 | no | gpio_sd5_out | 1'd1 |
| 106 | pwm1_f0_in | 0 | no | gpio_sd6_out | 1'd1 |
| 107 | pwm1_f1_in | 0 | no | gpio_sd7_out | 1'd1 |
| 108 | pwm1_f2_in | 0 | no | pwm1_out0a | 1'd1 |
| 109 | pwm0_cap0_in | 0 | no | pwm1_out0b | 1'd1 |
| 110 | pwm0_cap1_in | 0 | no | pwm1_out1a | 1'd1 |
| 111 | pwm0_cap2_in | 0 | no | pwm1_out1b | 1'd1 |
| 112 | pwm1_cap0_in | 0 | no | pwm1_out2a | 1'd1 |
| 113 | pwm1_cap1_in | 0 | no | pwm1_out2b | 1'd1 |
| 114 | pwm1_cap2_in | 0 | no | pwm2_out1h | 1'd1 |
| 115 | pwm2_flta | 1 | no | pwm2_out1l | 1'd1 |
| 116 | pwm2_fltb | 1 | no | pwm2_out2h | 1'd1 |
| 117 | pwm2_cap1_in | 0 | no | pwm2_out2l | 1'd1 |
| 118 | pwm2_cap2_in | 0 | no | pwm2_out3h | 1'd1 |
| 119 | pwm2_cap3_in | 0 | no | pwm2_out3l | 1'd1 |
| 120 | pwm3_flta | 1 | no | pwm2_out4h | 1'd1 |
| 121 | pwm3_fltb | 1 | no | pwm2_out4l | 1'd1 |
| 122 | pwm3_cap1_in | 0 | no | - | 1'd1 |
| 123 | pwm3_cap2_in | 0 | no | - | 1'd1 |
| 124 | pwm3_cap3_in | 0 | no | - | 1'd1 |
| 140 | I2S0I_DATA_in0 | 0 | no | I2S0O_DATA_out0 | 1'd1 |
| 141 | I2S0I_DATA_in1 | 0 | no | I2S0O_DATA_out1 | 1'd1 |
| 142 | I2S0I_DATA_in2 | 0 | no | I2S0O_DATA_out2 | 1'd1 |

| Signal No. | Input signals | Default value if unassigned | Same input signal from IO_MUX core | Output signals | Output enable of output signals |
|---|---|---|---|---|---|
| 143 | I2S0I_DATA_in3 | 0 | no | I2S0O_DATA_out3 | 1'd1 |
| 144 | I2S0I_DATA_in4 | 0 | no | I2S0O_DATA_out4 | 1'd1 |
| 145 | I2S0I_DATA_in5 | 0 | no | I2S0O_DATA_out5 | 1'd1 |
| 146 | I2S0I_DATA_in6 | 0 | no | I2S0O_DATA_out6 | 1'd1 |
| 147 | I2S0I_DATA_in7 | 0 | no | I2S0O_DATA_out7 | 1'd1 |
| 148 | I2S0I_DATA_in8 | 0 | no | I2S0O_DATA_out8 | 1'd1 |
| 149 | I2S0I_DATA_in9 | 0 | no | I2S0O_DATA_out9 | 1'd1 |
| 150 | I2S0I_DATA_in10 | 0 | no | I2S0O_DATA_out10 | 1'd1 |
| 151 | I2S0I_DATA_in11 | 0 | no | I2S0O_DATA_out11 | 1'd1 |
| 152 | I2S0I_DATA_in12 | 0 | no | I2S0O_DATA_out12 | 1'd1 |
| 153 | I2S0I_DATA_in13 | 0 | no | I2S0O_DATA_out13 | 1'd1 |
| 154 | I2S0I_DATA_in14 | 0 | no | I2S0O_DATA_out14 | 1'd1 |
| 155 | I2S0I_DATA_in15 | 0 | no | I2S0O_DATA_out15 | 1'd1 |
| 156 | - | - | - | I2S0O_DATA_out16 | 1'd1 |
| 157 | - | - | - | I2S0O_DATA_out17 | 1'd1 |
| 158 | - | - | - | I2S0O_DATA_out18 | 1'd1 |
| 159 | - | - | - | I2S0O_DATA_out19 | 1'd1 |
| 160 | - | - | - | I2S0O_DATA_out20 | 1'd1 |
| 161 | - | - | - | I2S0O_DATA_out21 | 1'd1 |
| 162 | - | - | - | I2S0O_DATA_out22 | 1'd1 |
| 163 | - | - | - | I2S0O_DATA_out23 | 1'd1 |
| 164 | I2S1I_BCK_in | 0 | no | I2S1I_BCK_out | 1'd1 |
| 165 | I2S1I_WS_in | 0 | no | I2S1I_WS_out | 1'd1 |
| 166 | I2S1I_DATA_in0 | 0 | no | I2S1O_DATA_out0 | 1'd1 |
| 167 | I2S1I_DATA_in1 | 0 | no | I2S1O_DATA_out1 | 1'd1 |
| 168 | I2S1I_DATA_in2 | 0 | no | I2S1O_DATA_out2 | 1'd1 |
| 169 | I2S1I_DATA_in3 | 0 | no | I2S1O_DATA_out3 | 1'd1 |
| 170 | I2S1I_DATA_in4 | 0 | no | I2S1O_DATA_out4 | 1'd1 |
| 171 | I2S1I_DATA_in5 | 0 | no | I2S1O_DATA_out5 | 1'd1 |
| 172 | I2S1I_DATA_in6 | 0 | no | I2S1O_DATA_out6 | 1'd1 |
| 173 | I2S1I_DATA_in7 | 0 | no | I2S1O_DATA_out7 | 1'd1 |
| 174 | I2S1I_DATA_in8 | 0 | no | I2S1O_DATA_out8 | 1'd1 |
| 175 | I2S1I_DATA_in9 | 0 | no | I2S1O_DATA_out9 | 1'd1 |
| 176 | I2S1I_DATA_in10 | 0 | no | I2S1O_DATA_out10 | 1'd1 |
| 177 | I2S1I_DATA_in11 | 0 | no | I2S1O_DATA_out11 | 1'd1 |
| 178 | I2S1I_DATA_in12 | 0 | no | I2S1O_DATA_out12 | 1'd1 |
| 179 | I2S1I_DATA_in13 | 0 | no | I2S1O_DATA_out13 | 1'd1 |
| 180 | I2S1I_DATA_in14 | 0 | no | I2S1O_DATA_out14 | 1'd1 |
| 181 | I2S1I_DATA_in15 | 0 | no | I2S1O_DATA_out15 | 1'd1 |
| 182 | - | - | - | I2S1O_DATA_out16 | 1'd1 |
| 183 | - | - | - | I2S1O_DATA_out17 | 1'd1 |

| Signal No. | Input signals | Default value if unassigned | Same input signal from IO_MUX core | Output signals | Output enable of output signals |
|---|---|---|---|---|---|
| 184 | - | - | - | I2S1O_DATA_out18 | 1'd1 |
| 185 | - | - | - | I2S1O_DATA_out19 | 1'd1 |
| 186 | - | - | - | I2S1O_DATA_out20 | 1'd1 |
| 187 | - | - | - | I2S1O_DATA_out21 | 1'd1 |
| 188 | - | - | - | I2S1O_DATA_out22 | 1'd1 |
| 189 | - | - | - | I2S1O_DATA_out23 | 1'd1 |
| 190 | I2S0I_H_SYNC | 0 | no | pwm3_out1h | 1'd1 |
| 191 | I2S0I_V_SYNC | 0 | no | pwm3_out1l | 1'd1 |
| 192 | I2S0I_H_ENABLE | 0 | no | pwm3_out2h | 1'd1 |
| 193 | I2S1I_H_SYNC | 0 | no | pwm3_out2l | 1'd1 |
| 194 | I2S1I_V_SYNC | 0 | no | pwm3_out3h | 1'd1 |
| 195 | I2S1I_H_ENABLE | 0 | no | pwm3_out3l | 1'd1 |
| 196 | - | - | - | pwm3_out4h | 1'd1 |
| 197 | - | - | - | pwm3_out4l | 1'd1 |
| 198 | U2RXD_in | 0 | yes | U2TXD_out | 1'd1 |
| 199 | U2CTS_in | 0 | yes | U2RTS_out | 1'd1 |
| 200 | emac_mdc_i | 0 | no | emac_mdc_o | emac_mdc_oe |
| 201 | emac_mdi_i | 0 | no | emac_mdo_o | emac_mdo_o_e |
| 202 | emac_crs_i | 0 | no | emac_crs_o | emac_crs_oe |
| 203 | emac_col_i | 0 | no | emac_col_o | emac_col_oe |
| 204 | pcmfsync_in | 0 | no | bt_audio0_irq | 1'd1 |
| 205 | pcmclk_in | 0 | no | bt_audio1_irq | 1'd1 |
| 206 | pcmdin | 0 | no | bt_audio2_irq | 1'd1 |
| 207 | - | - | - | ble_audio0_irq | 1'd1 |
| 208 | - | - | - | ble_audio1_irq | 1'd1 |
| 209 | - | - | - | ble_audio2_irq | 1'd1 |
| 210 | - | - | - | pcmfsync_out | pcmfsync_en |
| 211 | - | - | - | pcmclk_out | pcmclk_en |
| 212 | - | - | - | pcmdout | pcmdout_en |
| 213 | - | - | - | ble_audio_sync0_p | 1'd1 |
| 214 | - | - | - | ble_audio_sync1_p | 1'd1 |
| 215 | - | - | - | ble_audio_sync2_p | 1'd1 |
| 224 | - | - | - | sig_in_func224 | 1'd1 |
| 225 | - | - | - | sig_in_func225 | 1'd1 |
| 226 | - | - | - | sig_in_func226 | 1'd1 |
| 227 | - | - | - | sig_in_func227 | 1'd1 |
| 228 | - | - | - | sig_in_func228 | 1'd1 |

## A.3. Ethernet_MAC

**Table 20: Ethernet_MAC**

| PIN Name | Function6 | MII (int_osc) | MII (ext_osc) | RMII (int_osc) | RMII (ext_osc) |
|---|---|---|---|---|---|
| GPIO0 | EMAC_TX_CLK | TX_CLK (I) | TX_CLK (I) | CLK_OUT(O) | EXT_OSC_CLK(I) |
| GPIO5 | EMAC_RX_CLK | RX_CLK (I) | RX_CLK (I) | - | - |
| GPIO21 | EMAC_TX_EN | TX_EN(O) | TX_EN(O) | TX_EN(O) | TX_EN(O) |
| GPIO19 | EMAC_TXD0 | TXD[0](O) | TXD[0](O) | TXD[0](O) | TXD[0](O) |
| GPIO22 | EMAC_TXD1 | TXD[1](O) | TXD[1](O) | TXD[1](O) | TXD[1](O) |
| MTMS | EMAC_TXD2 | TXD[2](O) | TXD[2](O) | - | - |
| MTDI | EMAC_TXD3 | TXD[3](O) | TXD[3](O) | - | - |
| MTCK | EMAC_RX_ER | RX_ER(I) | RX_ER(I) | - | - |
| GPIO27 | EMAC_RX_DV | RX_DV(I) | RX_DV(I) | CRS_DV(I) | CRS_DV(I) |
| GPIO25 | EMAC_RXD0 | RXD[0](I) | RXD[0](I) | RXD[0](I) | RXD[0](I) |
| GPIO26 | EMAC_RXD1 | RXD[1](I) | RXD[1](I) | RXD[1](I) | RXD[1](I) |
| U0TXD | EMAC_RXD2 | RXD[2](I) | RXD[2](I) | - | - |
| MTDO | EMAC_RXD3 | RXD[3](I) | RXD[3](I) | - | - |
| GPIO16 | EMAC_CLK_OUT | CLK_OUT(O) | - | CLK_OUT(O) | - |
| GPIO17 | EMAC_CLK_OUT_180 | CLK_OUT_180(O) | - | CLK_OUT_180(O) | - |
| GPIO4 | EMAC_TX_ER | TX_ERR(O)* | TX_ERR(O)* | - | - |
| In GPIO Matrix* | - | MDC(O) | MDC(O) | MDC(O) | MDC(O) |
| In GPIO Matrix* | - | MDIO(IO) | MDIO(IO) | MDIO(IO) | MDIO(IO) |
| In GPIO Matrix* | - | CRS(I) | CRS(I) | - | - |
| In GPIO Matrix* | - | COL(I) | COL(I) | - | - |
| *Notes: 1. The GPIO Matrix can be any GPIO. 2. The TX_ERR (O) is optional. | | | | | |

## A.4. IO_MUX

For the list of IO_MUX pins please see the next page.

**IO_MUX**

| Pin No. | Power Supply Pin | Analog Pin | Digital Pin | Power Domain | Analog Function1 | Analog Function2 | Analog Function3 | RTC Function1 | RTC Function2 | Function1 | Type | Function2 | Type | Function3 | Type | Function4 | Type | Function5 | Type | Function6 | Type | Drive Strength (2'd2: 20 mA) | At Reset | After Reset |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | VDDA | | | VANA in | | | | | | | | | | | | | | | | | | | | |
| 2 | | LNA_IN | | VANA in | | | | | | | | | | | | | | | | | | | | |
| 3 | VDD3P3 | | | VANA in | | | | | | | | | | | | | | | | | | | | |
| 4 | VDD3P3 | | | VANA in | | | | | | | | | | | | | | | | | | | | |
| 5 | | SENSOR_VP | | VRTC | ADC_H | ADC1_CH0 | | RTC_GPIO0 | | GPIO36 | I | | | GPIO36 | I | | | | | | | | | ie=0 |
| 6 | | SENSOR_CAPP | | VRTC | ADC_H | ADC1_CH1 | | RTC_GPIO1 | | GPIO37 | I | | | GPIO37 | I | | | | | | | | | ie=0 |
| 7 | | SENSOR_CAPN | | VRTC | ADC_H | ADC1_CH2 | | RTC_GPIO2 | | GPIO38 | I | | | GPIO38 | I | | | | | | | | | ie=0 |
| 8 | | SENSOR_VN | | VRTC | ADC_H | ADC1_CH3 | | RTC_GPIO3 | | GPIO39 | I | | | GPIO39 | I | | | | | | | | | ie=0 |
| 9 | | CHIP_PU | | VRTC | | | | | | | | | | | | | | | | | | | | |
| 10 | | VDET_1 | | VRTC | | ADC1_CH6 | | RTC_GPIO4 | | GPIO34 | I | | | GPIO34 | I | | | | | | | | | ie=0 |
| 11 | | VDET_2 | | VRTC | | ADC1_CH7 | | RTC_GPIO5 | | GPIO35 | I | | | GPIO35 | I | | | | | | | | | ie=0 |
| 12 | | 32K_XP | | VRTC | XTAL_32K_P | ADC1_CH4 | TOUCH9 | RTC_GPIO9 | | GPIO32 | I/O/T | | | GPIO32 | I/O/T | | | | | | | 2'd2 | | ie=0 |
| | | | | | | | | | | | | | | | | | | | | | | | | |
| 13 | | 32K_XN | | VRTC | XTAL_32K_N | ADC1_CH5 | TOUCH8 | RTC_GPIO8 | | GPIO33 | I/O/T | | | GPIO33 | I/O/T | | | | | | | 2'd2 | | ie=0 |
| 14 | | | GPIO25 | VRTC | DAC_1 | ADC2_CH8 | | RTC_GPIO6 | | GPIO25 | I/O/T | | | GPIO25 | I/O/T | | | | | EMAC_RXD0 | I | 2'd2 | | ie=0 |
| 15 | | | GPIO26 | VRTC | DAC_2 | ADC2_CH9 | | RTC_GPIO7 | | GPIO26 | I/O/T | | | GPIO26 | I/O/T | | | | | EMAC_RXD1 | I | 2'd2 | | ie=0 |
| 16 | | | GPIO27 | VRTC | | ADC2_CH7 | TOUCH7 | RTC_GPIO17 | | GPIO27 | I/O/T | | | GPIO27 | I/O/T | | | | | EMAC_RX_DV | I | 2'd2 | | ie=1 |
| 17 | | | MTMS | VRTC | | ADC2_CH6 | TOUCH6 | RTC_GPIO16 | | MTMS | I0 | HSPICLK | I/O/T | GPIO14 | I/O/T | HS2_CLK | O | SD_CLK | I0 | EMAC_TXD2 | O | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 18 | | | MTDI | VRTC | | ADC2_CH5 | TOUCH5 | RTC_GPIO15 | | MTDI | I1 | HSPIQ | I/O/T | GPIO12 | I/O/T | HS2_DATA2 | I1/O/T | SD_DATA2 | I1/O/T | EMAC_TXD3 | O | 2'd2 | wpd, ie=1 | wpd, ie=1 |
| 19 | VDD3P3_RTC | | | VRTC supply in | | | | | | | | | | | | | | | | | | | | |
| 20 | | | MTCK | VRTC | | ADC2_CH4 | TOUCH4 | RTC_GPIO14 | | MTCK | I1 | HSPID | I/O/T | GPIO13 | I/O/T | HS2_DATA3 | I1/O/T | SD_DATA3 | I1/O/T | EMAC_RX_ER | I | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 21 | | | MTDO | VRTC | | ADC2_CH3 | TOUCH3 | RTC_GPIO13 | I2C_SDA | MTDO | O/T | HSPICS0 | I/O/T | GPIO15 | I/O/T | HS2_CMD | I1/O/T | SD_CMD | I1/O/T | EMAC_RXD3 | I | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 22 | | | GPIO2 | VRTC | | ADC2_CH2 | TOUCH2 | RTC_GPIO12 | I2C_SCL | GPIO2 | I/O/T | HSPIWP | I/O/T | GPIO2 | I/O/T | HS2_DATA0 | I1/O/T | SD_DATA0 | I1/O/T | | | 2'd2 | wpd, ie=1 | wpd, ie=1 |
| 23 | | | GPIO0 | VRTC | | ADC2_CH1 | TOUCH1 | RTC_GPIO11 | I2C_SDA | GPIO0 | I/O/T | CLK_OUT1 | O | GPIO0 | I/O/T | | | | | EMAC_TX_CLK | I | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 24 | | | GPIO4 | VRTC | | ADC2_CH0 | TOUCH0 | RTC_GPIO10 | I2C_SCL | GPIO4 | I/O/T | HSPIHD | I/O/T | GPIO4 | I/O/T | HS2_DATA1 | I1/O/T | SD_DATA1 | I1/O/T | EMAC_TX_ER | O | 2'd2 | wpd, ie=1 | wpd, ie=1 |
| | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | | | GPIO16 | VSDIO | | | | | | GPIO16 | I/O/T | | | GPIO16 | I/O/T | HS1_DATA4 | I1/O/T | U2RXD | I1 | EMAC_CLK_OUT | O | 2'd2 | | ie=1 |
| 26 | VDD_SDIO | | | VSDIO supply out/in | | | | | | | | | | | | | | | | | | | | |
| 27 | | | GPIO17 | VSDIO | | | | | | GPIO17 | I/O/T | | | GPIO17 | I/O/T | HS1_DATA5 | I1/O/T | U2TXD | O | EMAC_CLK_OUT_180 | O | 2'd2 | | ie=1 |
| 28 | | | SD_DATA_2 | VSDIO | | | | | | SD_DATA2 | I1/O/T | SPIHD | I/O/T | GPIO9 | I/O/T | HS1_DATA2 | I1/O/T | U1RXD | I1 | | | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 29 | | | SD_DATA_3 | VSDIO | | | | | | SD_DATA3 | I0/O/T | SPIWP | I/O/T | GPIO10 | I/O/T | HS1_DATA3 | I1/O/T | U1TXD | O | | | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 30 | | | SD_CMD | VSDIO | | | | | | SD_CMD | I1/O/T | SPICS0 | I/O/T | GPIO11 | I/O/T | HS1_CMD | I1/O/T | U1RTS | O | | | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 31 | | | SD_CLK | VSDIO | | | | | | SD_CLK | I0 | SPICLK | I/O/T | GPIO6 | I/O/T | HS1_CLK | O | U1CTS | I1 | | | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 32 | | | SD_DATA_0 | VSDIO | | | | | | SD_DATA0 | I1/O/T | SPIQ | I/O/T | GPIO7 | I/O/T | HS1_DATA0 | I1/O/T | U2RTS | O | | | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 33 | | | SD_DATA_1 | VSDIO | | | | | | SD_DATA1 | I1/O/T | SPID | I/O/T | GPIO8 | I/O/T | HS1_DATA1 | I1/O/T | U2CTS | I1 | | | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 34 | | | GPIO5 | VIO | | | | | | GPIO5 | I/O/T | VSPICS0 | I/O/T | GPIO5 | I/O/T | HS1_DATA6 | I1/O/T | | | EMAC_RX_CLK | I | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 35 | | | GPIO18 | VIO | | | | | | GPIO18 | I/O/T | VSPICLK | I/O/T | GPIO18 | I/O/T | HS1_DATA7 | I1/O/T | | | | | 2'd2 | | ie=1 |
| 36 | | | GPIO23 | VIO | | | | | | GPIO23 | I/O/T | VSPID | I/O/T | GPIO23 | I/O/T | HS1_STROBE | I0 | | | | | 2'd2 | | ie=1 |
| | | | | | | | | | | | | | | | | | | | | | | | | |
| 37 | VDD3P3_CPU | | | VIO supply in | | | | | | | | | | | | | | | | | | | | |
| 38 | | | GPIO19 | VIO | | | | | | GPIO19 | I/O/T | VSPIQ | I/O/T | GPIO19 | I/O/T | U0CTS | I1 | | | EMAC_TXD0 | O | 2'd2 | | ie=1 |
| 39 | | | GPIO22 | VIO | | | | | | GPIO22 | I/O/T | VSPIWP | I/O/T | GPIO22 | I/O/T | U0RTS | O | | | EMAC_TXD1 | O | 2'd2 | | ie=1 |
| 40 | | | U0RXD | VIO | | | | | | U0RXD | I1 | CLK_OUT2 | O | GPIO3 | I/O/T | | | | | | | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 41 | | | U0TXD | VIO | | | | | | U0TXD | O | CLK_OUT3 | O | GPIO1 | I/O/T | | | | | EMAC_RXD2 | I | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 42 | | | GPIO21 | VIO | | | | | | GPIO21 | I/O/T | VSPIHD | I/O/T | GPIO21 | I/O/T | | | | | EMAC_TX_EN | O | 2'd2 | | ie=1 |
| 43 | VDDA | | | VANA in | | | | | | | | | | | | | | | | | | | | |
| 44 | | XTAL_N | | VANA | | | | | | | | | | | | | | | | | | | | |
| 45 | | XTAL_P | | VANA | | | | | | | | | | | | | | | | | | | | |
| 46 | VDDA | | | VANA | | | | | | | | | | | | | | | | | | | | |
| 47 | | CAP2 | | VANA | | | | | | | | | | | | | | | | | | | | |
| 48 | | CAP1 | | VANA | | | | | | | | | | | | | | | | | | | | |
| Total Number | 8 | 14 | 26 | | | | | | | | | | | | | | | | | | | | | |

Note:
Please see Table: Notes on ESP32 Pin Lists.（请参考表：管脚清单说明。）