
ESP-FAQ

2020, Espressif Systems (Shanghai) PTE LTD

Oct 20, 2020

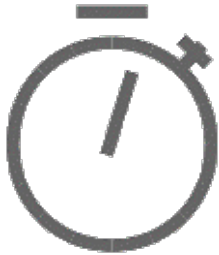
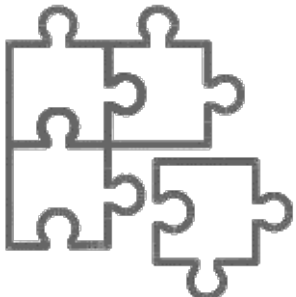
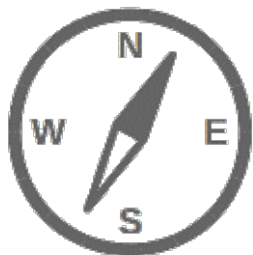



CONTENTS

1	Get started	3
1.1	Question search	3
1.2	Question feedback	3
1.3	Document contribution	3
2	Development environment	5
2.1	IDE plugins	5
2.2	Debugging	5
2.3	Environment setup	7
2.4	Firmware update	7
3	application solution	9
3.1	Artificial intelligence	9
3.2	AT application	9
3.3	Wifi mesh development framework	10
3.4	Ble mesh development framework	10
3.5	Audio development framework	10
3.6	Cloud service	10
3.7	Community sw and platforms	11
3.8	ios application	11
3.9	Android application	11
3.10	Camera application	11
4	Software framework	13
4.1	Wi-Fi	13
4.2	Bluetooth® LE & Bluetooth®	15
4.3	Ethernet	17
4.4	Peripherals	17
4.5	Protocols	18
4.6	Provisioning	19
4.7	Security	19
4.8	Storage	19
4.9	System	19
4.10	coexistence	21
5	Hardware related	23
5.1	Chip comparison	23
5.2	Development board	24
5.3	Hardware design	24
5.4	RF related	25

5.5	Process and ESD	25
5.6	Production test	25
6	Test verification	27
6.1	Power consumption verification	27
7	Commercial FAQ	29
7.1	Which certificates have your products passed?	29
7.2	Does your company have the ISO Quality Management System Certification?	29
7.3	Do your chips and modules have environmental certificates such as REACH, ROHS, etc?	29
7.4	Do you have distributors in China, Europe, the United States and Canada?	29
7.5	How can I start a distribution business with Espressif?	30
7.6	Where can I find your product information? Which of your products are in mass production?	30
7.7	Do your products have a longevity commitment?	30
7.8	Where can I find the SPQ (Standard Pack Quantity) and MOQ (Minimum Order Quantity) for your products?	30
7.9	What is your recommended purchasing method?	30
7.10	What's the price for bulk purchasing? How can I purchase in bulk?	30
7.11	Where can I find all the differences between your products (e.g. in terms of series and types)?	31
7.12	Do your products have firmware? Can I customize my module/chip flash before the product leaves the factory? How much does this process cost? How long does it take? How can you help me do this?	31
7.13	Which of your products support HomeKit? Where can I get the Espressif HomeKit SDK?	31
7.14	What is your company's address?	31
7.15	How can I contact your technical team?	32
7.16	How can I get in touch with your company?	32

□

ESP-FAQ is a summary document for common problems launched by Espressif.

		
Get Started	Development environment	Application solution
		
Software framework	Hardware related	Test verification

GET STARTED

□

1.1 Question search

□

1.2 Question feedback

□

1.3 Document contribution

□

Espressif Q & A.

DEVELOPMENT ENVIRONMENT

[]

2.1 IDE plugins

[]

2.2 Debugging

[]

2.2.1 How to troubleshoot in ESP32 Boot mode

By default, the boot information of ESP32-WROVER, which uses 1.8 V flash and psram, is 0x33 and 0x23 in Download mode. Besides, the boot information of other modules, which use 3.3 V flash and psram, is 0x13 and 0x03 in Download mode by default. for detailed information, please refer to Section Strapping Pins in [ESP32 Series Datasheet](#).

Thus, when ESP32 is started normally, its boot information should be 0x13, and the enabled pins are as the follows: - Pins: GPIO12GPIO0GPIO2GPIO4GPIO15GPIO5 - Levels: 0, 1, 0, 1, 0, 1

2.2.2 How can I make sending messages by UART0 disabled by default?

- For first-stage Bootloader log, connect GPIO15 to Ground.
- For second-stage Bootloader log, go to make `menuconfig > Bootloader config` to do configurations.
- For ESP-IDF log, go to make `menuconfig > Component config > Log output` to do some configurations.

2.2.3 How can I modify the default method of RF calibration in ESP32?

- During RF initialization, the partial-calibration method is used by default for RF calibration.
To use this method, please go to menuconfig and enable `CONFIG_ESP32_PHY_CALIBRATION_AND_DATA_STORAGE`.
 - If boot duration is not critical, please use the overall-calibration method instead.
To switch to the overall-calibration method, go to menuconfig and disable `CONFIG_ESP32_PHY_CALIBRATION_AND_DATA_STORAGE`.
 - It is recommended to use the **partial calibration** method, which ensures less boot time and enables you to add the function of triggering overall calibration as a last-resort remedy by erasing the NVS partition.
-

2.2.4 How can I modify the default method of RF calibration in ESP8266?

During RF initialization, the partial-calibration method is used by default for RF calibration. The initialization only takes little time, and for this method, the value of byte 115 in `esp_init_data_default.bin` is '0x01'. If boot duration is not critical, please use the overall-calibration method instead.

For NONOS SDK and versions of RTOS SDK earlier than 3.0, do either of the following:

- Call `system_phy_set_powerup_option(3)` in the function `user_pre_init` or `user_rf_pre_init`.
- In `phy_init_data.bin`, modify the value of byte 115 to '0x03'.

For RTOS SDK 3.0 and later versions:

- Go to menuconfig and disable `CONFIG_ESP_PHY_CALIBRATION_AND_DATA_STORAGE`.
- If `CONFIG_ESP_PHY_INIT_DATA_IN_PARTITION` is enabled in menuconfig, please modify the value of byte 115 in `phy_init_data.bin` to '0x03'; If `CONFIG_ESP_PHY_INIT_DATA_IN_PARTITION` is disabled, please modify the value of byte 115 in `phy_init_data.h` to '0x03'.

If you use the default method of RF calibration, and want to add the function of triggering overall calibration as a last-resort remedy:

- For NONOS SDK and versions of RTOS SDK earlier than 3.0: Please erase the RF parameters to trigger overall calibration.
 - For RTOS SDK 3.0 and later versions: Please erase the NVS partition to trigger overall calibration.
-

2.2.5 ESP8266 enters boot mode (2,7) and hits a watchdog reset. What could be wrong?

Please make sure that when ESP8266 boots, the strapping pins are held in the required logic levels. If externally connected peripherals drive the strapping pins to an inappropriate logic level, the ESP8266 may boot into an inappropriate mode of operation. In the absence of a valid program, the WDT may then reset the chip.

As good design practice, it is recommended that the strapping pins be used to interface to inputs of high impedance external devices only, which do not force the strapping pins high/ low during power-up. For more information, please refer to [ESP8266 Boot Mode Selection](#).

2.3 Environment setup

□

2.4 Firmware update

□

Espressif Q & A.

APPLICATION SOLUTION

[]

3.1 Artificial intelligence

[]

3.1.1 What kinds of cameras can be used on AI image recognition products?

With ESP32 as its main control chip, ESP-EYE supports various types of cameras, such as OV2640, OV3660, OV5640, OV7725 and etc (See [esp32-camera Github](#)).

3.1.2 Does esp-who support IDF 4.1?

No. Currently, the esp-who only supports IDF V3.3.1 and V 4.0.0 (esp-who commit: [2470e47 Update esp32-camera](#)). The subsequent supported versions will be updated on [ESP-WHO Github](#).

3.2 AT application

[]

3.2.1 Why is there a “no module named yaml” error when compiling ESP32-AT?

Please install the yaml module by using `python -m pip install pyyaml`.

3.2.2 Why are AT commands keep prompting “busy”?

The “busy” prompt indicates that the previous command has not been executed yet, and the system cannot respond to the current input. The processing mechanism of the AT commands is serial, i.e. one command at a time.

Any input through serial ports is considered to be a command input, so the system will also prompt “busy” or “ERROR” when there is any extra invisible character input.

For example, when users enter AT+GMR (line break CR LF) + (space) through serial ports, the system will execute the command immediately, because the AT+GMR (line break CR LF) is already considered to be a complete AT command.

Therefore, the space following the AT+GMR command will be treated as a second command. If AT+GMR has not been processed by the time of receiving the space, the system will prompt “busy”. However, if AT+GMR has been processed, the system will prompt “ERROR”, since space is an incorrect command.

3.2.3 Where can I get all the resources related to ESP32 AT?

- ESP32 AT bin files: <https://www.espressif.com/zh-hans/support/download/at>
- ESP32 AT document: [ESP32 AT Instruction Set and Examples](#)

You can also develop more AT commands based on the core codes of Espressif AT commands. Please find more information on ESP32 AT demos on GitHub: <https://github.com/espressif/esp-at>.

3.3 Wifi mesh development framework

□

3.4 Ble mesh development framework

□

3.5 Audio development framework

□

3.6 Cloud service

□

3.6.1 Are there any demo references for OTA upgrading?

- For ESP8266, please refer to [ESP8266 OTA](#).
- For ESP32 and ESP32-S2, please refer to [ESP32 and ESP32-S2 OTA](#).

3.7 Community sw and platforms

□

3.8 ios application

□

3.9 Android application

□

3.10 Camera application

□

Espressif Q & A.

SOFTWARE FRAMEWORK

□

4.1 Wi-Fi

□

4.1.1 Do ESP32 and ESP8266 support Chinese SSID for Wi-Fi?

Yes, but the CODEC format of router or smart phone should be the same.

For example, if both router and device use UTF-8 format, then the device can be successfully connected to the router with Chinese SSID.

4.1.2 Do Espressif's products support boundary scans?

No, they don't.

4.1.3 What is the definition for Wi-Fi channel? Can I select any channel of my choice?

A channel refers to a specific frequency channel within the allowable range of frequencies allocated for use by Wi-Fi systems. Different countries and regions use different channel numbers. Please refer to [ESP8266 Wi-Fi Channel Selection Guidelines](#).

4.1.4 What is the default network segment for ESP8266 SoftAP?

Why do I have problem connecting to router with IP 192.168.4.X in SoftAP + Station mode?

- The network segment used by ESP8266 SoftAP is 192.168.4.*, and its default IP address is 192.168.4.1. When connecting to router 192.168.4.X, errors may occur because the ESP8266 cannot distinguish if it should connect to the internal SoftAP or the external router.
-

4.1.5 How much time does an ESP32 scan take?

The total time for scanning depends on:

- Active scan (by default) or passive scan.
- The time spent on each channel is 120 ms for active scanning and 360 ms for passive scanning.
- The country code and configured channel range from 1~13 channels (by default).
- Fast scan (by default) or full-channel scan.
- Station mode or Station-AP mode, and if any active connections are currently maintained.

By default, channels 1 to 11 use active scans, and channels 12 to 13 use passive scans.

- In the absence of connection in Station mode, the total time for a full-channel scan is: $11*120 + 2*360 = 2040$ ms;
 - With active connections in Station mode or Station-AP mode, the total time for a full-channel scan is: $11*120 + 2*360 + 13*30 = 2430$ ms.
-

4.1.6 How can I optimize the second harmonic and other spurious signals created by my own products?

The second harmonic mainly comes from the radiation generated by the RF link and PA (power amplifier) power supply. The backplane (board size) and the product also make impact on the second harmonic. Therefore, it is recommended that you should:

- Add a ground capacitor of approximately 2.4 pF to the RF matching circuit to reduce the spurious radiation coming from the RF link;
 - Add a series inductor to the PA power supply (Pins 3 and 4 of ESP32) to reduce the spurious radiation coming from it.
-

4.1.7 How can I suppress the harmonics of 80 MHz?

If the harmonics of 80 MHz (160 MHz, 240 MHz, 320 MHz, etc) exceed the allowable levels of spurious emissions, you can add a resistor of approximately 470 Ω to the data transmission circuit (TXD) to suppress those harmonics.

4.1.8 What is the default IP address of ESP8266 SoftAP?

Why do I have problem connecting to router with IP 192.168.4.X in SoftAP + Station mode?

Default subnet of ESP8266 SoftAP is 192.168.4.*, and the default IP address is 192.168.4.1.

The ESP8266 cannot distinguish if the IP address of the SoftAP and the Station are both in the same subnet, in this case, 192.168.4.x.

4.2 Bluetooth® LE & Bluetooth®

□

4.2.1 Does ESP32 support Bluetooth® 5.0?

No, the ESP32 hardware only supports Bluetooth® LE 4.2.

The ESP32 has passed the Bluetooth® LE 5.0 certification, but some of its functions are still not supported on ESP32 (there will be a future chip which supports all functions in Bluetooth® LE 5.0).

4.2.2 Is it able to process OTA through Bluetooth on ESP32?

Yes, please operate basing on `bt_spp_acceptor` and `bt_spp_initiator` if using Bluetooth®; and basing on `ble_spp_server` and `ble_spp_client` is using Bluetooth® LE.

4.2.3 Could ESP32 realize bridging between Wi-Fi and Bluetooth® LE?

Yes, this function is developed on application layer. Users can retrieve data through Bluetooth® LE and send them out via Wi-Fi. For detailed information, please refer to [Wi-Fi and Bluetooth® LE Coexist demo](#).

4.2.4 After the Bluetooth® LE starts broadcasting, why some mobile phones cannot successfully scan broadcasts?

- Please check whether your mobile phone supports Bluetooth® LE function. Some mobile phones, such as iPhones, display Classic Bluetooth only in “Settings” -> “Bluetooth” (by default), and the Bluetooth® LE broadcast will be filtered out by the mobile phone.
 - It is recommended to use a dedicated Bluetooth® LE application to debug the Bluetooth® LE function. For example, LightBlue application can be used on iPhone.
 - Please check whether the broadcast packet conforms to the specified format. Mobile phones tend to filter out packets that do not conform to the specified format and display only the correct ones.
-

4.2.5 What is ESP32 Bluetooth® LE throughput?

- ESP32's Bluetooth® LE throughput depends on various factors such as environmental interference, connection interval, MTU size, and the performance of peer devices.
 - The maximum throughput of Bluetooth® LE communication between ESP32 boards can reach up to 700 Kbps, which is about 90 KB/s. For details, please refer to `ble_throughput` example in IDF.
-

4.2.6 How do ESP32 Bluetooth and Wi-Fi coexist?

In the menuconfig menu, there is a special option called “Software controls WiFi / Bluetooth coexistence”, which is used to control the ESP32's Bluetooth and Wi-Fi coexistence using software, thus balancing the coexistence requirement for controlling the RF module by both the Wi-Fi and Bluetooth modules. Please note that if `Software controls WiFi/Bluetooth coexistence` is enabled, the Bluetooth® LE scan interval shall not exceed `0x100 slots` (about 160 ms).

- If the Bluetooth® LE and Wi-Fi coexistence is required, this option can be enabled or disabled. However, if this option is not enabled, please note that the “Bluetooth® LE scan interval - Bluetooth® LE scan window” should be larger than 150 ms, and the Bluetooth® LE scan interval should be less than 500 ms.
 - If the Classic Bluetooth and Wi-Fi coexistence is required, it is recommended that you enable this option.
-

4.2.7 What is the transmit power of ESP32 Bluetooth®?

ESP32 Bluetooth has 9 transmit power levels, corresponding to -12 ~ 12 dBm, with a 3 dBm interval. The controller software limits the transmit power and selects the power level according to the corresponding power level declared by the product.

4.2.8 What is ESP32 Bluetooth® LE operating current?

Current	MAX (mA)	Min (mA)	Average (mA)
Advertising: Adv Interval = 40 ms	142.1	32	42.67
Scanning: Scan Interval = 160 ms, Window = 20 ms	142.1	32	44.4
Connection(Slave): Connection Interval = 20 ms, Iatency = 0	142.1	32	42.75
Connection(Slave): Connection Interval = 80 ms, Iatency = 0	142.1	32	35.33

4.2.9 What Bluetooth® LE profiles does ESP32 support?

At the moment, ESP32 Bluetooth® LE fully supports some basic profiles, such as GATT/SMP/GAP, and some self-defined profiles. The ones that have already been implemented include Bluetooth® LE HID (receiving side), Bluetooth® LE SPP-Like, Battery, DIS, Blu-Fi (Bluetooth Network Configuration-transmitting side), and so on.

4.2.10 How to connect mobile phones and play music using ESP32 Bluetooth®?

ESP32 is used as A2DP receiver when connected to a cell phone to play music.

Please note that the A2DP Sink Demo uses a mobile phone to obtain SBC encoded data stream only. In order to play sounds, you will also need to decode the data and some peripherals, including codec modules, D/A converter, and speaker.

4.2.11 What is the SPP performance of ESP32?

Using two ESP32 boards to run SPP, one-way throughput can reach up to 1900 Kbps (about 235 KB/s), which is close to the theoretical value in the specification.

4.3 Ethernet

□

4.4 Peripherals

□

4.4.1 Taken ESP-WROOM-S2 as the slave device and STM32 as MCU, is it possible to download through SPI interface?

No, we use UART0 to download by default. You can also design OTA support yourself in firmware.

4.4.2 What is the maximum speed supported by the SDIO interface?

The maximum clock speed supported by the hardware SDIO slave module is 50 MHz. As SDIO specifies use of quad data lines, the effective maximum bit rate is 200 Mbps.

4.4.3 Does the hardware SDIO interface support SD cards?

Please note that the SDIO hardware only supports the device or slave profile, i.e. it cannot act as a host to control SDIO devices such as SD cards.

4.5 Protocols

□

4.5.1 Does ESP8266 openssl support Hostname validation?

Yes. ESP8266 openssl is based on mbedTLS encapsulation, which supports Hostname validation. Use esp-tls to switch between mbedTLS and wolfSSL.

4.5.2 Does ESP8285 support CCS (Cisco Compatible eXtensions)?

No, it doesn't.

4.5.3 Does ESP8266 support HTTP hosting?

Yes, it does. ESP8266 can run as a server in both SoftAP and Station modes.

- When running as a server in SoftAP mode, clients can directly access the ESP8266 host or server at 192.168.4.1 (default) IP address.
 - When the server is accessed via a router, the IP address should be the one allocated to the ESP8266 by the router.
 - When using SDK to write native code, please refer to relevant examples.
 - When using AT commands, start a server using AT+CIPSERVER command.
-

4.5.4 How soon can the associated resources be released after the TCP connection is closed?

The associated resources can be released in 20 seconds or can be specified by the sent linger/send_timeout parameter.

4.6 Provisioning

□

4.7 Security

□

4.8 Storage

□

4.9 System

□

4.9.1 Is it possible to compile the binaries in application layer and bottom layer separately?

No, they cannot be compiled separately.

4.9.2 My application does not really need the watchdog timer, can I disable it?

The current SDK allows disabling the software watchdog only. The following methods can be taken to avoid watchdog reset when user program occupies CPU for too long:

- If your routine needs a time frame of duration between software reset and hardware watchdog reset, you may use `system_soft_wdt_stop ()` to disable the software watchdog. After the program has been executed, you can restart the software watchdog with `system_soft_wdt_restart ()`.
- You may feed the watchdog in between your codes by adding `system_soft_wdt_feed ()` so that the watchdog is updated before it issues a reset.

The hardware watchdog interrupt interval is 0.8×2048 ms, that is 1638.4 ms. The interrupt handling interval is 0.8×8192 ms, equal to 6553.6 ms. The interrupt handling interval is the time limit to feed the watchdog after the interrupt occurs. If the interrupt handling interval expires, it will trigger a hardware watchdog reset. As a result, in the cases where there is only hardware watchdog, if a program runs for over 6553.6 ms, then it could cause a hardware watchdog reset. If the program runs for over 8192 ms, then it will invoke a watchdog reset for sure.

The software watchdog is based on MAC timer and task arrangement. The interrupt interval is 1600 ms, so is the interrupt handling interval. As a result, in the cases where there are both software and hardware watchdogs, if a program runs for over 1600 ms, it could cause a software watchdog reset. If the program runs for over 3200 ms, it will invoke a watchdog reset for sure.

4.9.3 What are the differences between RTOS SDK and Non-OS SDK?

The main differences are as follows:

- Non-OS SDK
 - Non-OS SDK uses timers and callbacks as the main way to perform various functions - nested events and functions triggered by certain conditions. Non-OS SDK uses the espconn network interface; users need to develop their software according to the usage rules of the espconn interface.
- RTOS SDK
 1. FreeRTOS SDK is based on FreeRTOS , a multi-tasking OS. You can use the standard FreeRTOS interfaces to realize resource management, recycling operations, execution delay, inter-task messaging and synchronization, and other task-oriented process design approaches. For the specifics of interface methods, please refer to the official website of FreeRTOS or the book USING THE FreeRTOS REAL TIME KERNEL-A Practical Guide.
 2. The network operation interface in RTOS SDK is the standard lwIP API. RTOS SDK provides a package which enables BSD Socket API interface. Users can directly use the socket API to develop software applications; and port other applications from other platforms using socket API to ESP8266, effectively reducing the learning and development cost arising from platform switch.
 3. RTOS SDK introduces cJSON library whose functions make it easier to parse JSON packets.
 4. RTOS is compatible with Non-OS SDK in Wi-Fi interfaces, SmartConfig interfaces, Sniffer related interfaces, system interfaces, timer interface, FOTA interfaces and peripheral driver interfaces, but does not support the AT implementation.

4.9.4 Why do I get compile errors when using IRAM_ATTR in Non-OS SDK?

The default function attribute is IRAM_ATTR in Non-OS SDK. Therefore, if you want the function to reside in IRAM, please leave out the ICACHE_FLASH_ATTR attribution in the function definition/declaration.

4.9.5 Where is main function in ESP8266?

- ESP8266 SDK does not provide main function.
- Main function is stored in first-stage bootloader in ROM, which is used to load second-stage bootloader.
- The entry function of the second-stage bootloader is ets_main. After startup, the user_init in the user application will be loaded to lead the user to the program.

4.10 coexistence

□

Espressif Q & A.

HARDWARE RELATED

□

5.1 Chip comparison

□

5.1.1 What's the difference between single-core and dual-core of ESP32 (programming method, features performance, power consumption, and etc.)?

The main difference would be the additional independent core, on which some highly real-time operations can be located.

- The programming method is the same, and users only have to configure the freertos to run on the single core. See `make menuconfig` → Component config → FreeRTOS → [*] Run FreeRTOS only on first core;
- From the performance aspect, they are only different when it comes to high-load calculations. If not regarding to such calculations, there is no obvious difference in use (e.g., AI algorithm, high real-time interruption);
- There is only a slice of difference in power consumption when entering modem-sleep mode. For more details, please refer to [ESP32 Technical Reference Manual](#).

5.1.2 What's the difference between ESP32 E03 and the previous version in software use?

Not much differences in software use, and this version is compatible to old firmwares and some bugs in hardware have been solved. For more information, please refer to [ESP32 ECO V3 User Guide](#).

5.2 Development board

[]

5.3 Hardware design

[]

5.3.1 How to avoid the VDD3P3_RTC being powered down after ESP32 entering light-sleep mode?

After ESP32-WROVER-B entering light-sleep mode, the GPIO levels corresponding to pads powered by VDD3P3_RTC may be decreased. It is generally because of the power-down of RTC after entering light-sleep mode. Please call `esp_sleep_pd_config(ESP_PD_DOMAIN_RTC_PERIPH, ESP_PD_OPTION_ON)` to maintain the power supply of RTC.

5.3.2 The pins for I2S signals

The pins for I2S signals are located far apart from one another in the reference designs provided by Espressif. Can these pins be located closer together? For example, configure all the I2S signals to GPIO5, GPIO18, GPIO23, GPIO19 and GPIO22; and configure all the I2C signals to GPIO25 and GPIO26, or GPIO32 and GPIO33.

All I2S I/Os can be allocated freely. Please note that some I/Os can only be used as input pins. For details, please refer to the last page of [ESP32 Datasheet](#).

5.3.3 What are the general power supply requirements of the ESP8266?

- Digital voltage requirement: 1.8 V - 3.3 V
- Analog voltage requirement: 3.0 V - 3.6 V (The lowest possible analog voltage is 2.7 V.)
- Peak analog circuit current: 350 mA
- Peak digital circuit current: 200 mA

Note: CHIP_EN works at 3.0 V - 3.6 V, please use a level converter to ensure compatibility with digital logic at 1.8 V.

5.4 RF related

□

5.5 Process and ESD

□

5.6 Production test

□

Espressif Q & A.

TEST VERIFICATION

[]

6.1 Power consumption verification

[]

6.1.1 What sleep modes does ESP32 have? What is the difference between them?

ESP32 has three sleep modes: Modem-sleep, Light-sleep, and Deep-sleep.

- Modem-sleep:
 - The Station Legacy Fast sleep mode specified in the Wi-Fi specification, in which the Station sends a NULL frame to notify the AP to sleep or wake up.
 - After the station is connected to the AP, the station is automatically turned on. After the station enters the Modem-sleep mode, the RF module is shut down. During the modem sleep, the connection to the AP is maintained. After the AP disconnects from the station, Modem-sleep does not work.
 - After ESP32 enters Modem-sleep mode, the CPU clock frequency can be lowered to further reduce the current.
- Light-sleep:
 - A Station sleep mode based on Modem-sleep;
 - The differences between Light-sleep and Modem-sleep are:
 - * After ESP32 enters the Light-sleep mode, not only the RF module but also the CPU and part of the system clock are suspended.
 - * After ESP32 exits the Light-sleep mode, the CPU resumes working.
- Deep-sleep:
 - A sleep mode that is not specified in the Wi-Fi specification;
 - After ESP32 enters the Deep-sleep mode, all modules are closed except for RTC modules;
 - After ESP32 exits the Deep-sleep mode, the entire system reruns, which is similar to the system reboot;
 - During the deep sleep, no connection to the AP is maintained.

Espressif Q & A.

COMMERCIAL FAQ

□

7.1 Which certificates have your products passed?

Please check our [Certificates](#) , where you can get all the relevant information about our products.

7.2 Does your company have the ISO Quality Management System Certification?

Yes, our company has passed the ISO9001:2015 Quality Management System Certification.

7.3 Do your chips and modules have environmental certificates such as REACH, ROHS, etc?

Our chips and modules comply with REACH, ROHS, Prop65 and many other environmental certification standards. To find out more about them, please contact our business support team by submitting a [Sales Questions](#) electronic form, where you should mention the specific environmental certificate you need to check.

7.4 Do you have distributors in China, Europe, the United States and Canada?

To get specific information about our worldwide distributors, you should contact us by filling in the required information on our [Sales Questions](#) webpage. Then, our business support team will contact you as soon as possible, giving you all the information you need.

7.5 How can I start a distribution business with Espressif?

If you are interested in becoming one of our distributors, please send your company information to: dm@espressif.com.

7.6 Where can I find your product information? Which of your products are in mass production?

You can get the basic information on our products by clicking [here](#) . If you are looking for the technical documents of our products, please click [here](#) .

7.7 Do your products have a longevity commitment?

Yes, Espressif provides a minimum longevity commitment of at least 12 years for all the products listed [here](#) .

7.8 Where can I find the SPQ (Standard Pack Quantity) and MOQ (Minimum Order Quantity) for your products?

Please refer to our [Product Ordering Information](#) , where you can find our products' SPQ and MOQ.

7.9 What is your recommended purchasing method?

If you need to make a bulk purchase, please go to our [Sales Questions](#) and fill in the required information. Then, our business support team will contact you as soon as possible. If you just want to buy samples, please click [here](#) to check the corresponding purchasing method.

7.10 What's the price for bulk purchasing? How can I purchase in bulk?

Please go to our [Sales Questions](#) and fill in the electronic form you will find there. Then, our business support team will contact you as soon as possible.

7.11 Where can I find all the differences between your products (e.g. in terms of series and types)?

Please click [here](#) to find some introductory information on our products. For detailed information, please contact our sales team by clicking [here](#) .

7.12 Do your products have firmware? Can I customize my module/chip flash before the product leaves the factory? How much does this process cost? How long does it take? How can you help me do this?

Espressif Systems has developed a set of AT commands that can be used for Espressif products to easily interface with other products. Most of our modules have a standard AT firmware by default. For more information, please go to our [Sales Questions](#) and fill in the required details. Our business support team will contact you as soon as possible. Additionally, in order to simplify and shorten our customers' manufacturing process, we also provide customized manufacturing services. You can go to our [Manufacturing Services](#) and check all the available flash projects. For more information, please go to our [Sales Questions](#) and fill in the required details. Then, our business support team will get in touch with you as soon as possible.

7.13 Which of your products support HomeKit? Where can I get the Espressif HomeKit SDK?

You can refer to the [Espressif HomeKit SDK](#) . Please note that the Espressif HomeKit SDK is available to MFi licensees only, and you need to provide your Account Number (6 digits) for verification purposes, when [requesting the SDK](#) .

7.14 What is your company's address?

Espressif Systems (688018.SH) is a public multinational, fabless semiconductor company established in 2008, with headquarters in Shanghai and offices in Greater China, Singapore, India, the Czech Republic and Brazil. Please click [here](#) to check the specific addresses of Espressif's global offices.

7.15 How can I contact your technical team?

Please click [here](#) and tell us your problems or questions. We will try to help you as soon as possible.

7.16 How can I get in touch with your company?

In order to better understand your questions and needs, please click [here](#) and fill in the required information. Then, we will get in touch with you as soon as possible.

- [genindex](#)