ESP-FAQ

Mar 22, 2021

CONTENTS

[]

[**Coming Soon . . .**]

ESP-FAQ is a summary document of frequently asked questions released by Espressif. This repository aims to help our users to quickly locate those questions and get answers through simple explanations. The current categories of FAQ cover: development environment, application solution, software framework, hardware related and test verification.

Instruction	Development environment	Application solution
	AI	
Software framework	Hardware related	Test verification

CHAPTER

ONE

INSTRUCTION

[]

This section provides instructions for using ESP-FAQ. The purpose of "Question search" is to help you quickly understand the search methods and categories of this repository, so as to save much time for searching. In the meantime, you are welcomed to make contributions directly to ESP-FAQ, such as fixing bugs and adding new documents. For detailed information about this process, please go to "Document contribution".

1.1 Question search

[]

This instruction includes the following two parts

- Question search techniques
- Question classification framework

1.1.1 Question Search Techniques

Currently, there are mainly two searching techniques:

- Search keywords
- Exclude a specific keyword

Search key words

Extract keywords from your question and search for them, then the search results should list the best matches.

For example, if you expect to ask a question as `What is the Bluetooth LE Throughput for ESP32?`

Then just searching keywords such as `ESP32`, `BLE` and `throughput` should give you the result.

Exclude a Specific Keyword

Add a tag – into the search content in the format: `keyword -excluded keyword`. By doing so, the search results will not show the specific keyword you excluded.

For example, if you search `ESP32 -ble`, then any results with `ble` inside will not be shown.

1.1.2 Question Classification Categories

Once you have mastered the abovementioned `question search techniques`, you can use the categories in ESP-FAQ for reference to extract keywords for the questions you expect to ask and then search for them. The framework of ESP-FAQ categories is shown as follows:

ESP-FAQ

Development environment

[]

IDE plugins

[]

How to add ESP32 development board on Arduino IDE?

- For installation instructions of arduino-esp32, please refer to arduino-ide.
- For instructions on how to add development boards on Arduino IDE, please refer to arduino Cores.

Debugging

[]

When using ESP32 with my product, what is the reason for it not following up after a quick powered-down and then powered-on?

Scenario: 220 V to 5 V, and 5 V to 3.3 V power supply. The product failed when powered down and then powered on in 220 V voltage. The error log is as follows:

```
brownout detector was triggered.
rst:0xc(SW_CPU_RESET),boot:0x13(SPI_FAST_FLASH_BOOT) configsip:0,SPI
```

- The log conveys a message that the voltage has decreased to the threshold of triggering hardware watchdog during the quick powering-down process.
- The system did not enter bootloader due to the wrong powering-on timing. This can be resolved by force pulling-down chip_PU.

- For more detailed description about the powering-on and reset timing of ESP32, please refer to ESP32 Datasheet.

What is the serial port name of Wi-Fi devices

- In Windows system: COM*
- In the Linux subsystem of Windows 10: /dev/ttyS*
- In Linux system: /dev/ttyUSB*
- In macOS system: /dev/cu.usbserial-*

How to block debugging messages sent through UART0 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 configurations.

How to modify the default method of RF calibration in ESP32?

- During RF initialization, the partial calibration solution is used by default. Go to menuconfig and enable the `CONFIG_ESP32_PHY_CALIBRATION_AND_DATA_STORAGE` option.
- If the boot time is not critical, the full calibration solution can be used instead. Go to menuconfig and disable the `CONFIG_ESP32_PHY_CALIBRATION_AND_DATA_STORAGE` option.
- It is recommended to use the **partial calibration** solution, which ensures less boot time and enables you to add the function of erasing RF calibration information in NVS so as to trigger the full calibration operation.

How to modify the default method of RF calibration in ESP8266?

During RF initialization, the partial calibration solution is used by default. The initialization only takes little time. And for this method, the value of byte 115 in esp_init_data_default.bin is `0x01`. If the boot time is not critical, the full calibration solution can be used instead.

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

- Call system_phy_set_powerup_option(3) in 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, do either of the followings:

- 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 partial calibration solution, and want to add the function of triggering the full calibration operation:

- For NONOS SDK and earlier versions of RTOS SDK 3.0, please erase the RF parameters to trigger the full calibration operation.
- For RTOS SDK 3.0 and later versions, please erase the NVS partition to trigger the full calibration operation.

How to troubleshoot in ESP32 Boot mode

- The ESP32-WROVER uses 1.8 V flash and PSRAM, which are `0x33` by default in boot status and `0x23` in download mode.
- Other modules use 3.3 V flash and PSRAM, which are `0x13` by default and `0x03` in download mode.
- For detailed information, please refer to Section Strapping Pins in ESP32 Series Datasheet. Taken `0x13` as an example, the pins are as follows:

Pins	GPIO12	GPIO0	GPIO2	GPIO4	GPIO15	GPIO5
Level	0	1	0	0	1	1

When debugging with ESP32 JLINK, an ERROR occurs as: No Symbols For Freertos. How to resolve such issue?

First of all, such issue will not affect your actual operations. Then, you can still find solutions on the community here.

How to monitor the free space of the task stack?

The function `vTaskList()` can be used to print the available space of the task stack regularly.

Is it possible to use JTAG to debug with ESP32-S2

Yes. For detailed information, please refer to ESP32-S2 JATG Debugging.

How to modify the log output without changing the output level of menuconfig

The output level of log can be modified by using function `esp_log_level_set()`.

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.

When using the ESP-WROVER-KIT board with openocd, an error occurred as: Can't find board/esp32-wrover-kit-3.3v.cfg. How to resolve such issue?

- With 20190313 and 20190708 versions of openocd, please use instruction `openocd -f board/esp32-wrover.cfg`.
- With 20191114 and 20200420 (2020 later versions) versions of openocd, please use instruction `openocd -f board/esp32-wrover-kit-3.3v.cfg`.

How to obtain and parse coredump with ESP32?

- To obtain the 64 K coredump file from the whole firmware, you need to know its offset from the partition table. If we assume the offset is `0x3F0000`, the instruction should be as follows:

```
python esp-idf/components/esptool_py/esptool/esptool.py -p /dev/ttyUSB* read_
↪flash 0x3f0000 0x10000  coredump.bin
```

- Use the coredump reading script to convert the file obtained at the first step into readable messages. If we assume the coredump file is coredump.bin and the elf file is hello_wolrd.elf, the instruction should be as follows:

```
python esp-idf/components/espcoredump/espcoredump.py info_corefile -t raw -c␣
↪coredump.bin hello_world.elf
```

How to do RF performance test with ESP32&ESP8266&ESP32S2?

- Please refer to ESP32&ESP8266&ESP32S2 RF Performance Test Demonstration.

My PC cannot recognize the device connected in Win 10 system. What could be the reasons?

- Check if the device is identified in the Linux virtual subsystem of Win 10.
- If the device cannot be identified only in Win 10 system, go to Device Manager to see whether such device exists (e.g., COM x). If the answer is still no, please check your cable and driver.
- If the device cannot be identified only in Linux virtual subsystem, taken VMWare as an example, please go to "Settings" > "USB Controller" and select "Show all USB input devices".

One error occurred with ESP32 as: Core 1 paniced (Cache disabled but cache memory region accessed). What could be the reasons?

Reason:

- During the time when cache is disabled (e.g., when using the API spi_flash to read/write/erase/map the SPI flash), an interrupt is generated and the interrupt program accesses the flash resources.
- It is usually because the processor called programs from the flash and used its constants. One important thing is that since the Double variable is implemented through software, thus when this kind of variable is used in the interrupt programs, it is also implemented in the flash (e.g., forced type conversion operation).

Solution:

- Add an IRAM_ATTR modifier to the accessed function during interrupt
- Add an DRAM_ATTR modifier to the accessed constant during interrupt
- Do not use Double variable in the interrupt programs

How to read flash model information of the modules?

- Please use the python script esptool to read information of Espressif's chips and modules.

```
esptool.py --port /dev/ttyUSB* flash_id
```

What should I do when the Ethernet demo in debugging IDF has the following log

```
emac: Timed out waiting for PHY register 0x2 to have value 0x0243(mask␣
↪0xffff). Current value:
```

You can refer to the following configurations of the development board. Please see the schematics for details:

- CONFIG_PHY_USE_POWER_PIN=y
- CONFIG_PHY_POWER_PIN=5

Environment setup

[]

When setting up ESP32-S2 environment using command `idf.py set-target esp32-s2`, an error occurred as "Error: No such command 'set-target'". What could be the reason?

- The esp-idf is adapted to ESP32-S2 from release/v4.2, thus setting up ESP32-S2 environment in previous versions will cause errors. In this case, when using command `idf.py set-target esp32-s2`, there will be error as "Error: No such command 'set-target'".
- It is recommended to perform tests and development on ESP32-S2 using esp-idf release/v4.2 and later versions. For more information, please refer to ESP32-S2 Get Started.

When using idf.py menuconfig to build, how to deal with errors as "Configuring incomplete, errors occured"?

Please check your CMake version first using `camke --version`. If it is lower than version 3.10.0, please update your CMake version:

- Download CMake: https://CMake.org/download/.
- For details, please refer to http://www.mamicode.com/info-detail-2594302.html.

When installing esp-idf version master using ESP-IDF Tools 2.3 in Windows system, an error occurred as: Installation has failed with exit code 2. What could be the reason?

This is related to the bad network environment. The github repository cannot be downloaded smoothly under such network environment, causing the SDK failed downloading on your PC.

When setting up environment using esp-idf-tools-setup-2.3.exe in Windows system, errors occurred in "make menuconfig":

```
-- Warning: Did not find file Compiler/-ASM Configure
-- Configuring incomplete, erros occurred!
```

This is because the system could not find the compiling tool. You can test and verify this in example esp-idf/examples/get-started/hello_world.

When using esp-idf-tools-setup-2.2.exe in Windows system, a python error occurred during the installation:

```
Installation has failed with exit code 1
```

1. Update your tool chain: https://dl.espressif.com/dl/esp-idf-tools-setup-2.3.exe
2. Remove the obsolete option "–no-site-packages" from idf_tools.py

---

What should I do if I get `Download failed: security channel support error` when installing build environment in Windows system?

This is due to the Windows system has disabled the SSl3.0 support by default.

Solution: Go to *Control Panel* and find `Internet option`, select `Advanced` and check the `use SSL 3.0` option.

When executing export.bat in Windows system, what should I do if I get CMake, gdbgui version errors?

```
C:\Users\xxxx\.espressif\tools\cmake\3.16.4\bin
The following Python requirements are not satisfied:
gdbgui>=0.13.2.0
```

This is because the upstream gdbgui has been updated, thus it is not compatible with the low version of python. The current solution is to manually modify the root file `requirements.txt` in esp-idf by changing the description of gdbgui version to `gdbgui==0.13.2.0`.

Errors occurred when using idf.menuconfig and idf.build after updating the idf version from v3.3 to the latest one:

- Rebuild the environment following Get Started.
- Remove build and sdkconfig under the hello_world directory.

How to configure `PATH` and `IDF_PATH` when developing ESP32 and ESP8266 simultaneously?

- For `PATH`, there is no need to do extra configurations. You can put them together as: export PATH="$HOME/esp/xtensa-esp32-elf/bin:$HOME/esp/xtensa-lx106-elf/bin:$PATH".
- For `IDF_PATH`, you can specify it for separate chips as:

  In ESP32 related projects, use `IDF_PATH = $(HOME)/esp/esp-idf`; in ESP8266 related projects, use `IDF_PATH = $(HOME)/esp/ESP8266_RTOS_SDK`.

Firmware update

[]

How does the host MCU flash ESP32 via serial interfaces?

- For the related protocol, please refer to ESP32 Serial Protocol.
- For code examples, please refer to esp-serial-flasher.

How to automatically flash Espressif's modules using the USB-Serial tool?

The methods are as follows:

Modules	3V3	GND	TXD	RXD	IO0	EN
Serial tool	3V3	GND	RXD	TXD	DTR	RTS

Note: For ESP8266 modules, IO15 should be specially connected to ground.

How to flash firmware in macOS and Linux systems?

- For Apple system (macOS), you can use esptool downloaded via brew or git to flash firmware.
- For Linux system (e.g., ubuntu), you can use esptool downloaded via apt-get or git to flash firmware.

Does ESP32 support programming using JTAG pins directly?

Yes, please refer to JTAG Debugging.

Does ESP_Flash_Downloader_Tool support customized programming control

- The GUI tool is not open sourced and does not support embedded executive script.
- The low-level component ESPtool is open sourced and can be used to perform functions such as flashing and encryption. It is recommended to conduct secondary development based on this component.

Can I enable the Security Boot function for ESP32 via OTA?

- This is not recommended since it may cause risks and needs to upgrade OTA firmware for multiple times.
- Since the Security Boot function is in Bootloader, please update Bootloader first to enable this function.
  1. First, check the partition table of your current device to see if it can store the Bootloader with Security Boot function enabled.
  2. Then, update an intermediate firmware which can be written in Bootloader partition. By default, the Bootloader partition cannot be erased or written, you need to enable them via *make menuconfig*.
  3. Sign the intermediate firmware and upgrade it to the target device through OTA. Then upgrade the Bootloader of this firmware and the signed new firmware through OTA.
  4. If there are situations as powered-down or network break-down and restart during the Bootloader OTA process, the device cannot be started and needs to be re-flashed.

When flashing firmware to ESP32-S2, an error occurred as "A fatal error occurred: Invalid head of packet (0x50)". How to resolve such issue?

Scenario:

When flashing firmware to an ESP32-S2 device based on ESP-IDF v4.1, the following error code occurred:

```
esptool.py v2.9-dev
Serial port /dev/ttyUSB0
Connecting....
Chip is ESP32S2 Beta
Features: Engineering Sample
Crystal is 40MHz
MAC: 7c:df:a1:01:b7:64
Uploading stub...
Running stub...

A fatal error occurred: Invalid head of packet (0x50)
esptool.py failed with exit code 2
```

Solution

If the chip you are using is ESP32-S2, not ESP32-S2 Beta, please update ESP-IDF to v4.2 or later versions.

Notes:

- ESP-IDF v4.1 only supports ESP32-S2 Beta, which is not compatible with ESP32-S2.
- The version of esptool came with ESP-IDF v4.1 is v2.9-dev, which also only supports ESP32-S2 Beta.
- ESP-IDF v4.2 supports ESP32-S2 chips, and its esptool is v3.0-dev, which supports ESP32-S2 too.

How to download firmware based on esp-idf using flash_download_tool?

- Taken hello-world example for instance, please refer to get-started-guide when building an esp-idf project for the first time.
- Run `idf.py build` (Only for esp-idf v4.0 or later versions. Please use `make` for previous versions). After the build finished, the following flash command for the bin file will be generated:

```
#Project build complete. To flash, run this command:
../../../components/esptool_py/esptool/esptool.py -p (PORT) -b 921600 write_
↪flash --flash_mode dio --flash_size detect --flash_freq 40m 0x10000 build/
↪hello-world.bin  build 0x1000 build/bootloader/bootloader.bin 0x8000 build/
↪partition_table/partition-table.bin
or run 'idf.py -p PORT flash'
```

What is the communicationg protocol for flashing ESP chips?

- ESP Serial Protocol: Serial-Protocol.
- Python-based implementation: esptool.
- C-language based implementation: esp-serial-flasher.

application solution

[]

Artificial intelligence

[]

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 0V2640, OV3660, OV5640, OV7725 and etc (See esp32-camera Github).

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.

AT application

[]

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`.

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.

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.

Wifi mesh development framework

[]

Ble mesh development framework

[]

Audio development framework

[]

Cloud service

[]

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.

Community sw and platforms

[]

ios application

[]

Android application

[]

Camera application

[]

Software framework

[]

Bluetooth LE & Bluetooth

[]

When porting example gatt_server, an error occurred indicating head file does not exist. What could be the reasons?

When porting example gatt_server, an error occurred as `fatal error: esp_gap_ble_api.h: No such file or directory`, but this file is already included in the head file.

- Check sdkconfig to see whether sdkconfig.defaults is ported from the example or not. By default, Bluetooth® is disabled in SDK and needs to be enabled manually.
- If you are using cmake, the link configurations in the CMakeLists.txt file should be copied from the example too.

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).

After the Bluetooth® LE starts advertising, why some mobile phones cannot successfully scan them?

- 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 advertisement will be filtered out.
- 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 advertising 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.

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 LE, please operate basing on ble_spp_server and ble_spp_client.

How does ESP32 Bluetooth® and Bluetooth® LE dual-mode coexist and use?

The ESP32 Bluetooth and Bluetooth LE dual-mode does not require complex configurations. For developers, it is simple as calling Bluetooth LE API for Bluetooth LE, and calling Classic Bluetooth API for Classic Bluetooth.

For specifications on Classic Bluetooth and Bluetooth LE coexistence, please refer to ESP32 BT&BLE Dual-mode Bluetooth.

What is the throughput of ESP32 Bluetooth® LE?

- The throughput of ESP32 Bluetooth LE 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 example ble_throughput in ESP-IDF.

Does ESP32 support Bluetooth® 4.2 DLE (Data Length Extension)

Yes, Bluetooth 4.2 DLE is supported in all versions of ESP-IDF. There is no sample code provided currently. You can implement this by calling corresponding APIs directly. Please refer to esp_ble_gap_set_pkt_data_len.

How do ESP32 Bluetooth® and Wi-Fi coexist?

In the menuconfig, there is a special option called "Software controls WiFi/Bluetooth coexistence", which is used to control the coexistence of Bluetooth and Wi-Fi for ESP32 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 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 to enable this option.

How to get ESP32 Bluetooth® Compatibility Test Report?

Please contact sales@espressif.com.

What is the transmit power of ESP32 Bluetooth®?

The ESP32 Bluetooth has 9 transmit power levels, corresponding to -12 ~ 12 dBm of transmit power, 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.

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.

What is the operating current of ESP32 Bluetooth® LE?

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, latency = 0	142.1	32	42.75
Connection(Slave): Connection Interval = 80 ms, latency = 0	142.1	32	35.33

What kinds of Bluetooth® LE profiles does ESP32 support?

Currently, ESP32 Bluetooth LE fully supports some basic profiles, such as GATT/SMP/GAP, as well as 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.

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

ESP32 is used as an 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.

How is the ESP32 SPP performance?

When 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 specifications.

What is the maximum transmission rate for ESP32 Bluetooth® LE?

The transmission rate of ESP32 Bluetooth LE can reach 800 Kbps tested in a shielded box.

How does ESP32 Bluetooth® LE enter Light-sleep mode?

In hardware level, a 32 kHz external crystal oscillator should be added, or the Light-sleep mode will not take effect.

In software level (SDK4.0 and later versions), the following configurations should be enabled in menuconfig:

- Power Management:| menuconfig —> Component config —> Power management —>[*] Support for power management
- Tickless Idle:| menuconfig —> Component config —> FreeRTOS —>[*] Tickless idle support (3) Minimum number of ticks to enter sleep mode for (NEW)

Note: Tickless idle needs to be enabled to allow automatic light sleep. FreeRTOS will enter Light-sleep mode if no tasks need to run for 3 ticks (by default), that is 30 ms if tick rate is 100 Hz. Configure the FreeRTOS tick rate to be higher if you want to allow shorter duration light sleep, for example: menuconfig —> Component config —> FreeRTOS ->(1000) Tick rate (Hz).

- Configure external 32.768Hz crystal as RTC clock source :| menuconfig —> Component config —> ESP32-specific —>RTC clock source (External 32kHz crystal)[*] Additional current for external 32kHz crystal

Note: The “additional current” option is a workaround for a hardware issue on ESP32 that the crystal can fail in oscillating. Please enable this option when you use external 32 kHz crystal. This hardware issue will be resolved in the next ECO chip.

- Enable Bluetooth modem sleep with external 32.768kHz crystal as low power clock :| menuconfig —> Component config —> Bluetooth —> Bluetooth controller —> MODEM SLEEP Options —>[*] Bluetooth modem sleep

---

Are there any documentation references for ESP32 BluFi networking?

For BluFi networking, please refer to ESP32 Blufi. For BluFi networking examples, please refer to Blufi.

What is the maximum transmission rate for ESP32 Classic Bluetooth® SPP?

In an open environment, the transmission rate for ESP32 Classic Bluetooth SPP can reach 1400+ Kbps ~ 1590 Kbps (only for reference, please do tests based on your actual application environment) with bi-directional transmitting and receiving simultaneously.

Is ESP32 Bluetooth® compatible to Bluetooth® ver2.1 + EDR protocol?

Yes. The ESP32 Bluetooth is downward-compatible, you can do tests using our official Bluetooth examples.

How many Bluetooth® clients can be connected to ESP32?

The Bluetooth LE server supports up to nine client connections, please check the configuration of parameter ble_max_conn for applications. For stable connection, three clients should be good.

How to get the MAC address of Bluetooth® devices for ESP32?

You can get the MAC address configured by Bluetooth via API esp_bt_dev_get_address(void);, also the system pre-defined MAC address types via API esp_err_t esp_read_mac(uint8_t* mac,esp_mac_type_ttype);.

What is the default Bluetooth® transmit power for ESP32 SDK?

- By default, the power level of ESP32 SDK is 4, and the corresponding transmit power is 0 dBm.
- The power level of ESP32 Bluetooth ranges from 0 to 7, with the corresponding transmit power ranges from -12 dBm to 9 dBm. Each time the power level increases 1, the corresponding transmit power will increase by 3 dBm.

Is it possible to use Wi-Fi Smartconfig and Bluetooth® LE Mesh for ESP32 simultaneously?

It is not recommended to use them simultaneously.

- The Smartconfig will need to receive the networking data, thus occupying the antenna all the time. If it is used together with Bluetooth LE Mesh, there will be an extremely high rate of failure.
- The Bluetooth LE Mesh can be used together with Blufi. So it is recommended to use Blufi for networking.

What is the operating current for ESP32 Classic Bluetooth®

A2DP (Single core CPU 160 MhzDFS = falsecommit a7a90f)

Current	Maximum (mA)	Minimum (mA)	Average
Scanning	106.4	30.8	37.8
Sniff	107.6	31.1	32.2
Play Music	123	90.1	100.4

How to modify the transmit power for ESP32 Bluetooth®

The Bluetooth transmit power can be configured via function esp_ble_tx_power_set();. Please refer to esp_bt.h.

How is the networking compatibility of ESP32 Bluetooth® LE? Is it open-sourced?

- ESP32 Bluetooth networking, Blu-Fi networking for short, has a good compatibility as Bluetooth LE and is compatible with many mainstream mobile phones such as Apple, HUAWEI, Mi, OPPO, MEIZU, OnePlus, ZTE and etc.
- Currently, the Blu-Fi protocol and phone application code is not open-sourced.

When executing example bt_spp_acceptor on ESP32, the IOS device cannot find the ESP32 device during scanning. What could be the reasons?

- Apple has opened Bluetooth® as: A2DP, HID's keyboard, avrcp, SPP (need MFI), high-level Bluetooth LE and ANCS for Bluetooth LE.
- If the IOS device expects to communicate with the end device via SPP, the SPP of the end device should have the MFI certificate. However, ESP32 SPP does not have the MFI certificate, thus the IOS device cannot find ESP32 during scanning.

How is the security of ESP32 Bluetooth® LE/Bluetooth® Secure Simple Pairing (SSP) compared to legacy pairing?

- Secure Simple Pairing (SSP) is more secure than legacy pairing.
- The legacy pairing uses symmetric encryption algorithm, while Secure Simple Pairing (SSP) uses asymmetric cryptography algorithm.

How to certify the MTU size of ESP32 Bluetooth® LE?

- By default, the MTU size of ESP32 Bluetooth LE is 23 bytes, and can be configured to reach 517 bytes.
- For phones, the MTU size can be self-defined. Then, the end device with a smaller MTU will be chose for communication.

When advertising in ESP32 Bluetooth® LE mode, an error occurred as "W (17370) BT_BTM: data exceed max adv packet length". How to resolve such issue?

- This is because the advertising data has exceeded the maximum advertising packet length.
- The maximum data length of advertising payload is 31 bytes. If the actual data length exceeds 31 bytes, the Bluetooth protocol stack will drop some data and generate an error warning.
- If the data to be advertised exceeds the maximum packet length, the extra data can be put in the scan response packet.

Does ESP32 Bluetooth® LE support Client-Server mode, in which gatt server and gatt client can coexist?

- Yes, please refer to example gattc_gatts_coex.

What are the risks if there are over 6 devices connected to ESP32 Bluetooth® LE?

- Usually it depends on the specific application scenario. In general, the ESP32 Bluetooth LE can communicate stably with 3 devices connected.
- There is no exact number for maximum Bluetooth LE connections. When there are multiple devices connected to Bluetooth LE simultaneously, the RF is time-multiplexed, thus requiring the designer to ensure that each device is not overly occupied, causing other devices to timeout and disconnected.
- The connection parameters include: connection interval, connection window, latency and timeout. It is ok for devices to not respond within the `latency`, but if the responding time exceeds `timeout` threshold, the device will be disconnected.
- If the `interval` is configured to 100 and `window` to 5, the Bluetooth LE will be able to connect to more devices with Wi-Fi disconnected. However, If Wi-Fi is connected and the value of `interval` is too small, only a few devices can be connected.

- When the Bluetooth LE supports multiple devices connected simultaneously, there will be bigger possibility for RF solt management to generate error. So when there are multiple connections for Bluetooth LE, it is necessary to debug for different scenarios.

When using ESP32 device as the server of Bluetooth® LE, how many client devices can be connected?

- The ESP32 Bluetooth LE supports up to 9 client devices for connection. It is recommended to hold this number within 3.
- Please make configurations via menuconfig -> Component config -> Bluetooth -> Bluetooth controller -> BLE MAX Connections.

How to send files via Bluetooth® BR/EDR for ESP32?

- Please refer to example `bt_spp_acceptor` or `bt_spp_initiator` in classic bt.

When downloading example ESP_SPP_SERVER for ESP32, how to modify the name of the Bluetooth® device?

- The name of the Bluetooth device can be modified via `adv` parameter:

```
static const uint8_t spp_adv_data[23] = {
  0x02,0x01,0x06,
  0x03,0x03,0xF0,0xAB,
  0x0F,0x09,0x45,0x53,0x50,0x5f,0x53,0x50,0x50,0x5f,0x53,0x45,0x52,0x56,0x45,
↪0x52};
```

- The "0x0F" on the third line means the length of the following data is 15, "0x09" stands for data type (fixed) and data from "0x45" indicates the corresponding ASCII code of the device names (BLE_SPP_SERVER by default).

Ethernet

[]

When building an example on ESP32 Ethernet development board, an error occurred as "emac: Reset EMAC Timeout". What could be the reasons?

This is because the EMAC initialization is timeout, and is possibly related to the RMII clock. It is recommended to check your hardware, e.g., see if the PHY crystal oscillator is a cold joint.

When ESP32 connected to LAN8720 externally, with GPIO0 providing CLK, the initialization of Ethernet example failed. How to resolve such issue?

```
I (229) cpu_start: App cpu up.
I (247) heap_init: Initializing. RAM available for dynamic allocation:
I (254) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (260) heap_init: At 3FFB40A8 len 0002BF58 (175 KiB): DRAM
I (266) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (273) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (279) heap_init: At 400885D0 len 00017A30 (94 KiB): IRAM
I (285) cpu_start: Pro cpu start user code
I (303) cpu_start: Chip Revision: 1
W (303) cpu_start: Chip revision is higher than the one configured in␣
↪menuconfig. Suggest to upgrade it.
I (307) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
I (319) system_api: Base MAC address is not set, read default base MAC␣
↪address from BLK0 of EFUSE
E (1329) emac: Timed out waiting for PHY register 0x2 to have value␣
↪0x0007(mask 0xffff). Current value 0xffff
E (2329) emac: Timed out waiting for PHY register 0x3 to have value␣
↪0xc0f0(mask 0xfff0). Current value 0xffff
E (2329) emac: Initialise PHY device Timeout
ESP_ERROR_CHECK failed: esp_err_t 0xffffffff (ESP_FAIL) at 0x40084140
0x40084140: _esp_error_check_failed at /mnt/hgfs/workspace/esp32/IDF/esp-idf-
↪v3.3/components/esp32/panic.c:720

file: "/mnt/hgfs/workspace/esp32/project/ethernet/main/ethernet_example_main.
↪c" line 153
func: app_main
expression: esp_eth_enable()

ELF file SHA256:␣
↪``597d55ebf237c1cffa5f47c73148a159b22726d94a7b78100bd941d7d5fc906e``

Backtrace: 0x40083cdc:0x3ffb5e80 0x40084143:0x3ffb5ea0 0x400d32c1:0x3ffb5ec0␣
↪0x400d1742:0x3ffb5f20 0x40085d91:0x3ffb5f40
0x40083cdc: invoke_abort at /mnt/hgfs/workspace/esp32/IDF/esp-idf-v3.3/
↪components/esp32/panic.c:715

0x40084143: _esp_error_check_failed at /mnt/hgfs/workspace/esp32/IDF/esp-idf-
↪v3.3/components/esp32/panic.c:721

0x400d32c1: app_main at /mnt/hgfs/workspace/esp32/project/ethernet/main/
↪ethernet_example_main.c:153 (discriminator 1)

0x400d1742: main_task at /mnt/hgfs/workspace/esp32/IDF/esp-idf-v3.3/
↪components/esp32/cpu_start.c:542
```

(continues on next page)

(continued from previous page)

```
0x40085d91: vPortTaskWrapper at /mnt/hgfs/workspace/esp32/IDF/esp-idf-v3.3/
↪components/freertos/port.c:403


Rebooting...
```

- Please check if there are capacitors on IO0. It is better to have no capacitors on IO0 when it is used as the CLK output pin, or it would affect the timing.
- When GPIO0 starts to output RMII clock, please remember to check the `CONFIG_PHY_CLOCK_GPIO0_OUT` option in Kconfig.
- For references about Ethernet, besides the README in example, you can also refer to API Reference.

When using the Ethernet example in ESP-IDF, an error occurred as "Timed out waiting for PHY register 0x3 to have value 0xc0f0 (mask 0xfff0). Current value 0xffff". How to resolve such issue?

- Please refer to BBS issue and Github issue.
- When the value of the PHY register is 0xFFFF, please check the following:
  a. If the wiring of MDIO and MDC is correct
  b. If the 50 MHz clock required by RMII is normal
  c. If the address of PHY (both software and hardware) is correctly configured
- It is strongly recommended to check the strapping pins that control the PHY address to make sure they are not floating and **not in the default state**. Please make sure they have already been pulled up or down by external resistance.
- If you are still not sure about the PHY address, you can try to set the PHY address from 0 to 31, and read the PHY ID register to see if you can get the correct data. If yes, then write down the current PHY address.

When using ESP-IDF v4.1, how to set the static IP for ESP32 Ethernet?

Since v4.1 and later versions of ESP-IDF will remove the tcp/ip interface, it is recommended to use the ESP-NETIF interface.

Code example:

```
{
    ...
    esp_netif_config_t cfg = ESP_NETIF_DEFAULT_ETH();
    esp_netif_t *eth_netif = esp_netif_new(&cfg);
    // Set default handlers to process TCP/IP stuffs
    ESP_ERROR_CHECK(esp_eth_set_default_handlers(eth_netif));
    ...
    char* ip= "192.168.5.241";
    char* gateway = "192.168.5.1";
```

(continues on next page)

(continued from previous page)

```
    char* netmask = "255.255.255.0";
    esp_netif_ip_info_t info_t;
    memset(&info_t, 0, sizeof(esp_netif_ip_info_t));

    if (eth_netif)
    {
        ESP_ERROR_CHECK(esp_netif_dhcpc_stop(eth_netif));
        info_t.ip.addr = esp_ip4addr_aton((const char *)ip);
        info_t.netmask.addr = esp_ip4addr_aton((const char *)netmask);
        info_t.gw.addr = esp_ip4addr_aton((const char *)gateway);
        esp_netif_set_ip_info(eth_netif, &info_t);
    }
    ...
}
```

Is there any impact on Ethernet functionality if replacing the module of ESP32-Ethernet-Kit with ESP32-WROOM-32D?

- The ESP32-WROVER-B of ESP32-Ethernet-Kit can be replaced with ESP32-WROOM-32D, and its Ethernet functionality will not be affected.
- The main difference between ESP32-WROOM and ESP32-WROVER module series: ESP32-WROVER modules have a 4 MB PSRAM while ESP32-WROOM modules do not have any PSRAM by default. Please refer to:
  - ESP32-WROOM-32D Datasheet
  - ESP32-WROVER-B Datasheet
- The ESP32-WROOM and ESP32-WROVER modules all use the ESP32 chip as their core, which supports Ethernet. For more information, please refer to ESP32 Datasheet.
- Related document: ESP32-Ethernet-Kit Getting Started Guide.

coexistence

[]

When Wi-Fi coexists with ESP-BLE-MESH, what mode does it support?

For now, only Wi-Fi STA mode supports coexistence.

When Wi-Fi coexists with ESP-BLE-MESH, the Wi-Fi throughput is low, why?

For ESP32-DevKitC boards without PSRAM, Wi-Fi can coexist with ESP-BLE-MESH but with a relatively low throughput. For ESP32-DevKitC boards with PSRAM, the transmit rate can stabilize at over 1 Mbps.

To support PSRAM, the following configurations in menuconfig should be enabled accordingly:

- `ESP32-specific --> Support for external,SPI-connected RAM --> Try to allocate memories of Wi-Fi and LWIP...`
- `Bluetooth --> Bluedriod Enable --> BT/BLE will first malloc the memory from the PSRAM`
- `Bluetooth --> Bluedriod Enable --> Use dynamic memory allocation in BT/BLE stack.`
- `Bluetooth --> Blutooth controller --> BLE full scan feature supported.`
- `Wi-Fi --> Software controls Wi-Fi/Bluetooth coexistence --> Wi-Fi`

Does ESP32 support coexistence between 16 MB External Flash and 8 MB External PSRAM?

Yes, ESP32 supports coexistence between 16 MB External Flash and 8 MB External PSRAM.

Does ESP32 support coexistence between Wi-Fi Mesh and Bluetooth® LE Mesh?

No.

However, the ESP32 supports coexistence between Wi-Fi Mesh and Bluetooth LE, or Wi-Fi STA and Bluetooth LE Mesh.

Does ESP32 support coexistence between Bluetooth® and Wi-Fi?

Yes, but time-sharing control is required for ESP32's coexistence between Wi-Fi and Bluetooth. Please go to menuconfig to enable the Wi-Fi/Bluetooth coexistence, shown as follows:

```
menuconfig -> Component config -> Wi-Fi -> Software controls WiFi/
Bluetooth coexistence (Enable)
```

When Bluetooth® LE and A2DP coexist, audio data reception is lost and lagged while entering Bluetooth LE scanning. How to resolve such issue?

- Reduce the duty cycle of Bluetooth LE scanning
- Use RingBuf to cache audio data

Does ESP32 support coexistence between the network port (LAN8720) and Wi-Fi (Wifi-AP)?

Yes, this can be achieved by writing the detection events of both connections as one.

Peripherals

[]

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.

When using ESP32 to develop Touch Sensor applications, where can I find references?

Please refer to Software and Hardware Designs.

Is ESP-WROOM-02D module able to connect SPI flash?

The ESP-WROOM-02D has free SPI peripherals, and can be externally connected to SPI flash to store data.

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.

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.

Does ESP8266 support I2C slave mode?

No. If you want to use this function, it is recommended to choose ESP32 or ESP32-S2 chips instead. For ESP32 examples, please refer to i2C_self_test.

What should I pay attention to for ESP32 pin configurations?

The ESP32 has ESP32-WROOM and ESP32-WROVER series modules. Please pay attention to the following configurations with GPIOs.

The WROOM-32/32D/32U series have 26 pins available for customers. Please note:

- GPIO6 ~ GPIO11 are used by the internal flash and cannot be used elsewhere;
- GPIO34, 35, 36 and 39 are input-only pins and cannot be used for outputs;
- The ESP32 has a built-in GPIO Matrix, and some peripheral interfaces can be connected to any free pins. That is, for hardware designs, there is no need to strictly distribute some functions on certain pins.

For detailed information, please refer to Table 9 in ESP32 Datasheet.

The WROVERWROVER-IWROVER-BWROVER-IB series have 24 pins available for customers. Please note:

- GPIO6 ~ GPIO11 are used by the internal flash and cannot be used elsewhere;
- GPIO34, 35, 36 and 39 are input-only pins and cannot be used for outputs;
- For WROVER series, it is not recommended to use GPIO12 for Touch Sensor functions since it has been pulled up in the module;
- The ESP32 has a built-in Matrix, and some peripheral interfaces can be connected to any free pins. That is, for hardware designs, there is no need to strictly distribute some functions on certain pins.

For detailed information, please refer to Table 9 in ESP32 Datasheet.

There are three sets of UARTs in ESP32, but only UART0 can be used for downloading with fixed pins.

Does ESP32 support transmitting audio stream using A2DP?

Yes, please refer to example a2dp_source.

Is ESP8266 I2C realized via software programming?

Yes, ESP8266 I2C is realized via GPIO software programming.

When using ESP8266 NonOS v3.0 SDK, the following error occurred. What could be the reasons?

```
E:M 536    E:M 1528
```

Any error logs beginning with E:M indicate insufficient memory.

What is the frequency range for ESP8266 PWM?

The PWM of ESP8266 is realized via software programming, so the maximum CLK value is 1 M limited by timer. It is recommended to set the frequency to 1 K. The PWM frequency can also be improved by decreasing the resolution of duty cycle.

Are there any limits on outputting PWM via ESP32 GPIO pins?

The ESP32 can output PWM using any GPIO switched via IO Matrix except for GPIO34 ~ GPIO39, which are used for input only.

When there is water on ESP32-S2 Touch Sensor, does it block or recognize the Touch event with its waterproof function?

When the impact of water on the Touch Sensor is small (with droplets), the sensor will adapt to it actively; when the impact of water on the Touch Sensor is large (with large water flow), the sensor can avoid certain extent of the impact by configuring software to lock some sensor channels.

While the waterproof feature of ESP32-S2 Touch Sensor shielding the Touchpad with water flow, does other pads with no water still usable?

Yes, the specific shielding channel can be selected via software.

Are there any recommendations for materials that can be used to test Touch Sensor, can trigger Touch Sensor stably and is close to the parameters of human touches

For experiments with high consistency requirements, it is doable to replace human hands with cell phone pencils.

Can the pins of Touch Sensor be remapped

No, because Touch Sensor is realized via software programming.

Do I need to reset a check threshold for Touch Sensor after covering it with a acrylic plate

Yes.

Is it possible for Touch Sensor to detect whether there is a acrylic plate on the top, so that it can switch to the pre-defined threshold value automatically when there is a acrylic plate added or removed?

For now, it cannot adapt to the impacts brought by physical changes.

What is the maximum capacity for ESP32 SD card?

- In the SD3.01 Specifications, the SDXC card supports a maximum capacity of 2 TB (2048 GB).
- The ESP32 SDMMC Host also complies with the SD3.01 Specifications, which means up to 2 TB areas of it can be accessed by peripherals. When accessing the card via SPI bus using the SDSPI driver, there are also 2 TB of areas can be accessed in hardware level.
- In software level, the usable area of the card is also affected by the file system.

Does ESP32 support USB function?

- No, ESP32 does not support USB function.
- However, ESP32-S2 supports USB1.1.

What should I pay attention to when using the HW timer interrupt with ESP8266?

- Please refer to ESP8266 Technical Reference Manual regarding the related APIs.
- If you are using NonOS SDK, please refer to ESP8266 Non-OS SDK API Reference.
- Generally, when using hardware interrupts, you should finish executions as soon as possible and put the callback function into IRAM to avoid the potential impacts of Cache.
  - For RTOS SDK, IRAM_ATTR should be added to the function.
  - For NonOS SDK, ICACHE_FLASH_ATTR should not be added before the function.

Can I distribute the ESP32 PWM to any I/O?

- Theoretically, the PWM can be distributed to any I/Os except for those that only have input functions (e.g., GPIO34 ~ GPIO39).
- In the actual use, this could also be affected by the limitations of chips and modules, the un-pinned I/Os, flash occupations and etc.

Is there any example code for I2S driving LCD with ESP32?

Please refer to I2S LCD Driveresp-iot-solution i2s_devices.

When using ESP8266 RTOS SDK v2.1 and previous versions, how to set LOG to UART1?

After initializing UART1, you can switch LOG to UART1 via API:

```
UART_SetPrintPort(UART1);
```

When using ESP8266 RTOS SDK v3.0 and later versions, how to set LOG to UART1?

Go to `menuconfig -> Component config -> ESP8266-specific -> UART for console output -> custom -> UART peripheral to use for console output -> UART0` and change the option to "UART1".

How to enable UART Flow Control in ESP32 IDF?

- Hardware enable: uart-flow-control.
- Software enable: software-flow-control.

The PWM of ESP8266 NonOS SDK changes slow. What could be the reasons?

- If you are using the gradient APIs in SDK example/IOT_demo, e.g., light_set_aim or light_set_aim_r, it will need a gradual process for PWM changes.
- If you need the PWM Duty to take effect immediately after configuration, please call API pwm_set_duty, and call pwm_start next to make this configuration take effect.

Some ESP8266 GPIOs are high level. What could be the reasons?

- According to the hardware design, some GPIOs are pulled up or down by default. Thus the level of these pins are not controlled by the program during system initialization, causing some incorrect levels of GPIOs during the boot process.
- If you expect to use these GPIOs, it is recommended to keep the hardware peripherals be consistent with the default level status, or adjust level status in software during bootloader process. When using the later method, you may also encounter temporary level exception.

How is the accuracy of ESP8266 ADC?

- The ESP8266 ADC is 10 bit, and its theoretical accuracy is 2^{10} = 1024.
- After connected to a router, the ESP8266 will enter Modem-sleep mode from STA mode, causing the change of the reference value inside the chip. Therefore, the ADC could measure the data change.
- If you expect an accurate result, please read the ADC value using function system_adc_fast_read after turning off Wi-Fi.

How to get the Bitmap information of the ADC register?

Since the ADC of ESP8266 is highly integrated with the internal RF circuit, the Bitmap and register information is not opened. Please contact sales@espressif.com if you have any special needs.

How many channels does ESP32 ADC have? What is the sampling rate and significant digit

- The ESP32 ADC has 18 channels.
- Its sampling rate can reach 100000 times per second without Wi-Fi.
- Its sampling rate can reach 1000 times per second with Wi-Fi.
- The internal significant digit of ADC is 12-bit.

Can I disable the thread scheduling and use a single CPU for ESP32 to realize real-time GPIO?

- For now, we do not have any related configurations for SDK to support the single operation of CPU1. Both cores of ESP32 support SMP only, but not AMP.
- The following solutions can be used to resolve the issue of output waveform being interrupted:
  - Use hardware signal outputs, and choose related digital protocols to realize SPI, I2C, I2S and etc. For special usage with SPI, you can generate waveform using signal output lines.
  - See if the hardware RMT can generate the desired waveform with enough length.
  - When the hardware interrupt generated corresponding waveform, all callbacks need to be put in IRAM.
  - Use the co-processor in the chip as a single chip without an operation system. But it only supports assembly language for now.

Is there any reference for ESP32 Touch application?

For ESP32 Touch application, please refer to Touch Software and Hardware Designs.

Is it possible to use ESP32 SD card together with flash & PSRAM?

- Yes, they can be used simultaneously.
- However, they do not share the same group of SDIO.

When using UART0 as a serial communication port for ESP32, what should I pay attention to?

- Generally, it is not recommended to use UART0 as a normal serial communication port, because it is the default LOG output port.
- If the UART number in ESP32 is not enough for you or it is not convenient to change your hardware designs anymore, and UART0 is therefore going to be used as a normal communication port, please pay attention to the following suggestions:

Software:

You need to protect the serial communication port from being affected by printing. The UART0 mainly has three print settings in the default program:

- First, power-on ROM print. You can set the MTDO pin as low level when powered on to block the power-on ROM print.
- Second, bootloader log output. You can set `menuconfig -> Bootloader config -> Bootloader log verbosity` as `Not output` to block bootloader log output.
- Third, app log output. You can set `menuconfig -> Component config -> Log output -> Default log verbosity` as `Not output` to block app log output.

Hardware

- Pay attention to other devices on UART0 when downloading programs since they could affect downloading. It is recommended to reserve a 0 resistance between ESP32 and other devices so that if there is something wrong while downloading, you can still disconnect this resistance.

Is it possible to use GPIO34 GPIO39 from ESP32-SOLO-1 as the RX signal pin for UART and TWAI®?

Yes, GPIO34 GPIO39 are for receive only and can be used as the RX signal pins for UART and TWAI®.

Does ESP-WROOM-S2 module support using SDIO as a slave

Yes, because ESP-WROOM-S2 flash uses SPI interfaces.

Does ESP32 support using crystal oscillator as the clock source of I2S

No. Please go to ESP32 Technical Reference Manual to read about clock source configurations of I2S.

When calling the API adc_read_fast() with ESP8266, will it cause a Wi-Fi disconnection?

- Please turn off Wi-Fi and interrupts first before calling adc_read_fast(). Please refer to the Specification of this API.
- Since the API adc_read_fast() performs continuous acquisition and the ADC is partially coupled internally with Wi-Fi RF, so it is not possible to call this function with Wi-Fi turned on.
- Please use adc_read() for ADC acquisition when Wi-Fi is on. To ensure data stability, you need to use function esp_wifi_set_ps(WIFI_PS_NONE); to turn off Wi-Fi Modem-sleep mode.

Note: ADC sampling rate: can reach 100000 times per second with Wi-Fi turned off, and 1000 times per second with Wi-Fi turned on.

How to dynamically change the serial baud rate and make it take effect immediately with ESP32?

Please use the API uart_set_baudrate() to change the baud rate of UART. Please see API Reference.

Since ESP32-S2 has removed the SDIO interface, does it still support external TF card?

The ESP32-S2 has four groups of SPI interfaces, and you can use the interface of SPI2/SPI3 to connect an external TF card. When doing so, the SPI should be set to general SPI mode.

What is the turning speed of ESP32 GPIO levels?

It will take around 300 ns.

How to connect MIC with ESP32?

- You can connect I2S peripheral if it is digital MIC.
- You can connect ADC peripheral if it is analog MIC.

Does ESP32 support analog audio output or digital audio output?

- The ESP32 supports DAC analog audio output for simple outputs such as warning tones. But if you use it for music playing, the effect will not be so desirable.
- The ESP32 also supports I2S digital audio output. For I2S configurable pins, please see Section four in ESP32 Datasheet.

What is the difference of SPI, HSPI and VSPI in ESP32?

- The SPI/HSPI/VSPI in the *parallel QSPI* interface are groups to connect the external flash, which is mounted on the SPI group.
- Any usable HSPI/VSPI in the driver are general-purpose SPIs. The difference in their names are only used to distinguish between groups.

When certain RTC peripheralsSARADC1SARADC2AMPHALL are powered on, the inputs of GPIO36 and GPIO39 will be pulled down for approximately 80 ns.

When enabling power for any of these peripherals, ignore input from GPIO36 and GPIO39.

When the LEDC is in decremental fade mode, a duty overflow error can occur.

When using LEDC, avoid the concurrence of following three cases:

- The LEDC is in decremental fade mode;
- The scale register is set to 1;
- The duty is $2^{\text{LEDC\_HSTIMERx\_DUTY\_RES}}$ or $2^{\text{LEDC\_LSTIMERx\_DUTY\_RES}}$.

When the TWAI® controller enters reset mode or when the TWAI controller undergoes bus-off recovery, the REC is still permitted to change. How to resolve such issue?

When entering reset mode, the TWAI controller should set the the LISTEN_ONLY_MODE to freeze the REC. The desired mode of operation should be restored before exiting reset mode or when bus-off recovery completes.

When the TWAI® controller undergoes the bus-off recovery process, the controller must monitor 128 occurrences of the bus free signal before it can become error active again. How to resolve such issue?

When undergoing bus-off recovery, an error warning interrupt does not necessarily indicate the completion of recovery. Users should check the `STATUS_NODE_BUS_OFF` bit to verify whether bus-off recovery has completed.

Upon completion of bus-off recovery, the next message that the TWAI® controller transmits may be erroneous?

Upon detecting the completion of bus-off recovery (via the error warning interrupt), the CAN controller should enter then exit reset mode so that the controller's internal signals are reset.

When the TWAI® Controller receives an erroneous data frame, the data bytes of the next received data frame become invalid, how to resolve such issue?

Users can detect the errata triggering condition (i.e., bit or stuff error in the data or CRC field) by setting `INTERRUPT_BUS_ERR_INT_ENA` and checking `ERROR_CODE_CAPTURE_REG` when a bus error interrupt occurs. If the errata condition is met, the following workarounds are possible:

- The TWAI controller can transmit a dummy frame with 0 data byte to reset the controller's internal signals. It is advisable to select an ID for the dummy frame that can be filtered out by all nodes on the TWAI bus.
- Hardware reset the TWAI controller (will require saving and restoring the current register values).

The ESP32 GPIO peripheral may not trigger interrupts correctly if multiple GPIO pads are configured with edge-triggered interrupts. How to resolve such issue?

- Workaround 1:
  - Follow the steps below to trigger a GPIO interrupt on a rising edge:
    1. Set the GPIO interrupt type to high.
    2. Set the interrupt trigger type of the CPU to edge.
    3. After the CPU services the interrupt, change the GPIO interrupt type to low. A second interrupt occurs at this time, and the CPU needs to ignore the interrupt service routine.
  - Similarly, follow the steps below to trigger a GPIO interrupt on a falling edge:
    1. Set the GPIO interrupt type to low.
    2. Set the interrupt trigger type of the CPU to edge.
    3. After the CPU services the interrupt, change the GPIO interrupt type to high. A second interrupt occurs at this time, and the CPU needs to ignore the interrupt service routine.
- Workaround 2:

  Assuming GPIO0 ~ GPIO31 is Group1 and GPIO32 ~ GPIO39 is Group2.

  - If an edge-triggered interrupt is configured in either group then no other GPIO interrupt of any type should be configured in the same group.
  - Any number of level-triggered interrupts can be configured in a single group, if no edge-triggered interrupts are configured in that group.

Does ESP8266 support pulse counting?

- The ESP8266 does not include a hardware pulse counting module, thus only supports counting via the interrupt of GPIO rising edge or falling edge.
- When Wi-Fi is turned on in ESP8266, it may cause a vacuum in the GPIO sampling due to its high priority, thus interrupting the collected counts and causing data loss.
- In conclusion, it is recommended to use ESP32 and subsequent chips for scenarios with high counting demands.

Protocols

[]

Does ESP8266 OpenSSL support Hostname validation?

Yes. ESP8266 OpenSSL is based on mbedTLS encapsulation, which supports `Hostname validation`. ESP-TLS can be used to switch between mbedTLS and wolfSSL.

Does ESP32 support PCI-E protocol?

No, it doesn't.

How to optimize communication latency for ESP32

- It is recommended to turn off the sleep function for Wi-Fi by calling the API `esp_wifi_set_ps(WIFI_PS_NONE)`.
- You can also disable the `AMPDU` function in menuconfig.

Does ESP8285 support CCS (Cisco Compatible eXtensions)?

No, it doesn't.

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.

Does ESP32 support LoRa (Long Range Radio) communication?

No, the ESP32 itself does not have the LoRa protocol stack and the corresponding RF parts. However, to realize communication between Wi-Fi and LoRa devices, you can connect an external chip integrated with LoRa protocol to ESP32. In this way, ESP32 can be used as the master control MCU to connect the LoRa chip.

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.

How to configure the server address so as to make it an autonomic cloud platform by using MQTT?

Please refer to MQTT Examples.

With ESP32, are there any return instructions if I skip to a function using the `jump` instruction in ULP

Please see here for ULP CPU instructions list and corresponding specifications. Normally, a general register is used for return instructions to store backup PC addresses for later jumping backs. Since there are only four general registers in ULP for now, please make proper use of them.

After the SNTP calibration for ESP8266 RTOS SDK v3.2, errors gradually increase. How to resolve such issue

This is because the ESP8266 uses software timer, which brings large errors itself. You can improve it with the following solutions:

- For branch v3.2, you can resynchronize time (300 s is recommended) from the server regularly by creating a task.
- For branch release-v3.3, the code of system timer has been refactored and is tested with low errors. On the other hand, you can still synchronize time from the server regularly.
- The master branch has inherited the refactored code from branch release-v3.3. In addition, you can configure the SNTP synchronization interval in menuconfig: `Component config > LWIP > SNTP -> Request interval to update time (ms).`

Does ESP8266 support loop-back for device-end UDP broadcasts?

- Yes, it does.
- Please enable the LOOPBACK option from LWIP in menuconfig: `menuconfig -> Component config -> Enable per-interface loopback (type "Y" to enable).`

What is the default packet length for TCP/IP?

In default configurations, the single packet TCP is 1460 bytes and UDP is 1472 bytes.

When using UTC and GMT methods in SNTP protocol, why can't I get the time of the target time zone

- The "TZ = UTC-8" refers to POSIX time, in which "UTC" is the abbreviation of any time zone and the number is the number of hours that the time zone is behind UTC.
- "UTC-8" indicates a certain time zone, "UTC" for short, which is -8 hours later than the actual UTC. Therefore, "UTC+8" is 8 hours later than the actual UTC, and also 16 hours later than Beijing.

Is there any special firmware or SDK in ESP32 that can only provide AP/STA (TCP/IP bypass) without using its internal TCP/IP so as to give developers more permissions?

The software solution ESP-Dongle can fit your requirements. Please contact Business Team to sign NDA and then get related solutions.

Can I add any broadcast data I want to Android ESP-Touch (e.g., add a device ID so that ESP32 can receive this ID)?

- No, the data content sent under the current ESP-Touch protocol is fixed and cannot be customized.
- If you expect to send customized data, it is recommended to use Blufi, which is the networking protocol based on Bluetooth LE. Please refer to the following references for Blufi:
  - Android APPhttps://github.com/EspressifApp/EspBlufiForAndroid.
  - iOS APPhttps://github.com/EspressifApp/EspBlufiForiOS.

When testing RTOS-SDK mqtt/ssl_mutual_auth with ESP8266, the server connection failed. Why?

- The failure of SSL connection may due to insufficient memory of ESP8266.
- Please use the Master version of ESP8266-RTOS-SDK to test this example, since it supports dynamic memory allocation in menuconfig so as to reduce the usage of memory peak. The specific action is:
  - menuconfig -> Component config -> mbadTLS -> (type "Y" to enable) Using dynamic TX /RX buffer -> (type "Y" to enable) Free SSL peer certificate after its usage -> (type "Y" to enable) Free certificate, key and DHM data after its usage.

Provisioning

[]

Security

[]

Is the firmware in ESP8266 readable?

Yes, because the firmware in ESP8266 is located in the external flash, thus can be read externally. In addition, ESP8266 does not support flash encryption and all the data is written in plaintext.

Is it possible to encrypt firmware for ESP8285?

- No, the ESP8285 chip does not support firmware encryption function.
- Both ESP32 and ESP32-S2 support firmware encryption, thus can be your substitution.
- If you insist on using ESP8285, you can achieve data encryption by adding an encrypted chip externally.

After enabling secure boot, there is a build error indicating missing files. What could be the reasons

Error log: /d/ESP32/esp-mdf/esp-idf/components/bootloader_support/Makefile.projbuild:7/f/ESP32Root/secure_boot_signing_ke

Reason: security boot is a function for firmware signature verification, which requires generating a pair of certificates. For detailed information, please refer to Secure Boot.

After enabling secure boot, is it possible for modules to be flashed again?

- If the secure boot is configured as one-time, then it can only be flashed once and the bootloader firmware cannot be reflashed.
- If the secure boot is configured as reflashable, then the bootloader firmware can be flashed again.

With flash encrypted enabled, a module has an error as `flash read error` after reflashed. How to resolve such issue?

With flash encrypted enabled, the module will not support plaintext firmware flash. You can reflash it after disabling the encryption function via espefuse.py, or used the encryption key to flash ciphertext.

Note: Please note there is a time limit for flash encrypted function: it can only be enabled and disabled for 3 times.

After enabling flash encryption and secure boot for ESP32, how to disable them?

- If you are using the one-time flash (Release) mode, both flash encryption and secure boot cannot be disabled.
- If you are using the reflashable (Development (NOT SECURE)) mode, the flash encryption can be disabled, please refer to Disabling Flash Encryption; while the secure boot cannot be disabled.

Is there any security strategy for ESP32 to protect its firmware?

- ESP32 supports flash encryption and secure boot.
- For flash encryption, please refer to flash encryption.
- For secure boot, please refer to secure boot.
- For secure boot V2, please refer to ECO3 Chip secure boot V2.

When ESP32 debugging GDB after enabling flash encryption, why does it always continue to reset and restart?

- After ESP32 enabling flash encryption or secure boot, it will restrict JTAG debugging by default, please refer to Tips and Quirks.
- You can read the current JTAG status of your chip using the `espefuse.py summary` command from esptool.

Storage

[]

What is the requirement for the storage and usage of ESP32 flash?

The external flash can be mapped into CPU instruction space and RO data space simultaneously. ESP32 can support up to 16 MB of external flash.

- When it is mapped into CPU instruction space, up to 11 MB + 248 KB of data can be mapped at a time. If more than 3 MB + 248 KB is mapped at a time, the cache performance may be degraded due to speculative CPU reads.
- When it is mapped into RO data space, up to 4 MB of data can be mapped at a time, and 8-bit, 16-bit and 32-bit reads are supported.

When using ESP32 modules, how to check the size of their PSRAM?

First of all, you need to enable the PSRAM function in `make menuconfig`. Then, you can check its size via the log information of bootloader or by calling esp_spiram_get_size().

When ESP32 connected to a PSRAM externally, how to change its clock source?

In menuconfig: menuconfig -> Component config -> ESP32-specific -> SPI RAM config.

Is it possible to use ESP8266 together with TF card?

It is not recommended to use ESP8266 together with TF card.

- Although ESP8266 can be connected to TF card in hardware level (communicate through SPI), the chip may run out of memory in different application scenarios due to its limited resources. Thus, it is not recommended to use ESP8266 with TF card.
- If all you need is a Wi-Fi-only module that can be connected to a TF card, it is recommended to use the ESP32-S2 chip instead.

If data needs to be stored or updated to flash every minute, can ESP32 NVS meet this requirement?

According to NVS Specifications, NVS uses two main entities in its operation: pages and entries. Logical page corresponds to one physical sector of flash memory. For now, we assume that flash sector is 4096 bytes and each page can contain 126 entries (32 bytes for one entry), with the left spaces for page header (32 bytes) and entry state bitmap (32 bytes). Typical flash lifetime is 100 k erase cycles. Assuming that the device is expected to run for 10 years, and the data size written to flash is 4 bytes per minute with flash encryption disabled, then the number of flash write operation can be calculated as: 60×24×365×10=5256000. In this way, no more than 42 k of erase cycles (5256000/126) will be caused in NVS, which does not exceed 100 k. Therefore, such operation is supported even without the effects of multiple sectors. In actual use, usually there will be multiple sectors given to NVS, and the NVS can distribute erase cycles to different sectors, making the number of erase cycles in each sector necessarily less than 42 k.

Therefore, the NVS of ESP32 can meet such requirement.

What kind of sectors are reserved for customized use in ESP8266 modules?

- For previous versions of SDK rel3.0, besides for bootloader and app bin, the following sectors are reserved at the end of the configured flash: 1 for system information, 1 for OTA information and 1 for RF calibration information.
- For SDK rel3.0 and later versions, we use partition_table to manage flash. Except for partition_table and bootloader, other bin files are all marked in partition_table.

Does NVS have wear levelling function?

NVS does not use the wear_levelling component in ESP-IDF, but uses an erase levelling mechanism implemented internally. The flash in use is in a wear-levelling state.

Can NVS sectors be corrupted by accidental power loss during writing?

No, NVS is designed to resist accidental power loss, so it will not be damaged.

Can ESP32 mount a file system partition in the external SPI flash

Yes, this function has been added in ESP-IDF v4.0 and later versions. Please note that when two partitions are mounted to ESP32, it is not permitted for multiple tasks to write files into the same partition at the same time.

How to improve the damage to FATFS file system caused by accidental power loss?

Since FATFS is designed to not support write transactions, the accidental power loss during the erase process will cause error to partitions, which cannot be restored by simply modifying FATFS. For now, it is recommended to resolve this problem in application level by creating two identical FATFS partitions to do backups, or you can also choose a more secure file system instead, such as LittleFS and SafeFAT (charged).

How to make and flash the image of a FATFS file system?

Here we will use a third-party tool, since there is no such tool provided in ESP-IDF now. The entire process shows as below:

- Step 1: use the mkfatfs tool to create image in a specified folder. Here we create a 1048576-byte image named fat_img.bin in the file_image folder.

```
./mkfatfs -c file_image -s 1048576 ./fat_img.bin
```

- Step 2: flash the image to address 0x110000:

```
esptool.py -p /dev/ttyUSB1 -b 460800 --before default_reset --after hard_
↪reset write_flash --flash_mode dio --flash_size detect --flash_freq 80m␣
↪0x110000 ~/Desktop/fat_img.bin
```

- Step 3: mount the image in program:

```
static void initialize_filesystem() {
  static wl_handle_t
  wl_handle = WL_INVALID_HANDLE;
  const esp_vfs_fat_mount_config_t
  mount_config = { .max_files = 10, };
  ESP_LOGI(TAG, "Mounting FATfilesystem");
  esp_err_t err = esp_vfs_fat_spiflash_mount("/spiflash", "storage", &mount_
↪config, &wl_handle);
  if (err != ESP_OK) {
      ESP_LOGE(TAG, "Failed to mount FATFS (%s)", esp_err_to_name(err));
      return;
  }
}
```

Note: The address to be flashed in step 2 must be the corresponding partition address in the partition table where FATFS is mounted, and the image created must be the same size as the one set in the partition table. Please remember to go to menuconfig and set `Component config -> Wear Levelling -> Wear Levelling library sector size` to 512, or the mounting would fail.

Can ESP32 use LittleFS file system?

Currently, LittleFS is not included in ESP-IDF, but there is a third-party porting component esp_littlefs that can be used directly in ESP-IDF. You can use the mklittlefs tool for the image of LittleFS file system.

How to check the memory usage (e.g., DRAM, IRAM, rodata) of ESP32 chips?

You can check the usage of related memories for ESP32 chips by inputting the instruction *size-components* under corresponding directories in terminal, e.g., *make size-components* or *idf.py size-components*.

How to read flash data for ESP8266?

- You can use the script tool under ESP8266-RTOS-SDK to read flash data. The whole process is shown as follows:
  - Install python environment and the required packages;
  - Go to ESP8266_RTOS_SDK/components/esptool_py/esptool;
  - Run `python esptool.py --chip esp8266 --port /dev/ttyUSB0 --baud 115200 read_flash 0x0 0x400000 esp8266.bin`. In this command, "esp8266.bin" is a self-defined file, where all flash data read will be stored; "/dev/ttyUSB0" is the serial port number in linux environment, which can be different in other environments and systems.

When a 8 MB PSRAM mounted on ESP32, why only 4 MB of it is actually mapped?

- Up to 4 MB (0x3F80_0000 ~ 0x3FBF_FFFF) of external RAM can be mapped into data address space, please refer to the specifications of Section 3.1.4 Memory Map in ESP32 Datasheet.
- You can access the other 4 MB following example himem.

System

[]

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

No, they cannot be compiled separately.

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.

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.

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.

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.

Wi-Fi

[]

What is the one-to-one bit rate for ESP32 in ESP-NOW mode

Test result:

- Test board: ESP32_Core_board_V2.
- Wi-Fi mode: station.
- Around 214 Kbps in opened environment.
- Around 555 Kbps in shielding box.

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.

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.

[Scan] Do Espressif's products support boundary scans?

No, they don't.

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.

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.

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 to the data transmission circuit (TXD) to suppress those harmonics.

[LWIP] With ESP-IDF v4.1, how to configure ESP32's IP address when it is in SoftAP mode?

Since ESP-IDF v4.1 and later versions do not have TCP/IP interfaces anymore, it is recommended to use the ESP-NETIF interface instead.

Code example

```
{
    ...
    esp_netif_t *ap_netif = esp_netif_create_default_wifi_ap();
    char* ip= "192.168.5.241";
    char* gateway = "192.168.5.1";
    char* netmask = "255.255.255.0";
    esp_netif_ip_info_t info_t;
    memset(&info_t, 0, sizeof(esp_netif_ip_info_t));

    if (ap_netif)
    {
        ESP_ERROR_CHECK(esp_netif_dhcps_stop(ap_netif));
        info_t.ip.addr = esp_ip4addr_aton((const char *)ip);
        info_t.netmask.addr = esp_ip4addr_aton((const char *)netmask);
        info_t.gw.addr = esp_ip4addr_aton((const char *)gateway);
        esp_netif_set_ip_info(ap_netif, &info_t);
        ESP_ERROR_CHECK(esp_netif_dhcps_start(ap_netif));
    }
    ...
}
```

[LWIP] How to configure ESP32's static IP when it is in Station mode

Since ESP-IDF v4.2 and later versions do not have tcp/ip interfaces anymore, it is recommended to use the ESP-NETIF interface instead. The code example is as follows

```
esp_netif_ip_info_t info_t = {0};
esp_netif_dns_info_t dns_info = {0};

// Initialize TCP/IP network interface (should be called only once in␣
↪application)
ESP_ERROR_CHECK(esp_netif_init());
// Create default event loop that running in background
ESP_ERROR_CHECK(esp_event_loop_create_default());
esp_netif_config_t cfg = ESP_NETIF_DEFAULT_ETH();
esp_netif_t *eth_netif = esp_netif_new(&cfg);
// Set default handlers to process TCP/IP stuffs
ESP_ERROR_CHECK(esp_eth_set_default_handlers(eth_netif));
```

(continues on next page)

(continued from previous page)

```
esp_netif_dhcpc_stop(eth_netif);

info_t.ip.addr = ESP_IP4TOADDR(192,168,3,23);
info_t.gw.addr = ESP_IP4TOADDR(192,168,3,1);
info_t.netmask.addr = ESP_IP4TOADDR(255,255,255,0);
esp_netif_set_ip_info(eth_netif,&info_t);

dns_info.ip.u_addr.ip4.addr = ESP_IP4TOADDR(8,8,8,8);
esp_netif_set_dns_info(eth_netif,ESP_NETIF_DNS_MAIN,&dns_info);
```

[LWIP] How to configure the Option contents of DHCP Server in ESP-IDF?

Since ESP-IDF v4.1 and later versions do not have TCP/IP interfaces anymore, it is recommended to use the ESP-NETIF interface instead. You can also refer to this example when dealing with DHCP Client configuration. The code example is as follows:

```
// Set up the handle for softap netif
esp_netif_t *ap_netif = esp_netif_create_default_wifi_ap();

// ESP_NETIF_IP_ADDRESS_LEASE_TIME, DHCP Option 51, Set the lease time for␣
↪distributed IP address
uint32_t dhcps_lease_time = 60; // The unit is min
ESP_ERROR_CHECK(esp_netif_dhcps_option(ap_netif,ESP_NETIF_OP_SET,ESP_NETIF_
↪IP_ADDRESS_LEASE_TIME,&dhcps_lease_time,sizeof(dhcps_lease_time)));

// ESP_NETIF_DOMAIN_NAME_SERVER , DHCP Option 6, Set DNS SERVER
// Set the local domain DNS first
esp_netif_dns_info_t dns_info = {0};
dns_info.ip.u_addr.ip4.addr = ESP_IP4TOADDR(8,8,8,8);
ESP_ERROR_CHECK(esp_netif_set_dns_info(ap_netif,ESP_NETIF_DNS_MAIN,&dns_
↪info));

uint8_t dns_offer = 1; // Pass 1 to make the modified DNS take effect, if it␣
↪is 0, then it means the gw ip of softap is used as the DNS server (0 by␣
↪default)
ESP_ERROR_CHECK(esp_netif_dhcps_option(ap_netif,ESP_NETIF_OP_SET,ESP_NETIF_
↪DOMAIN_NAME_SERVER,&dns_offer,sizeof(dns_offer)));

// ESP_NETIF_ROUTER_SOLICITATION_ADDRESS, DHCP Option 3 Router, Pass 0 to␣
↪make the DHCP Option 3(Router) un-shown (1 by default)
uint8_t router_enable = 0;
ESP_ERROR_CHECK(esp_netif_dhcps_option(ap_netif,ESP_NETIF_OP_SET,ESP_NETIF_
↪ROUTER_SOLICITATION_ADDRESS,&router_enable, sizeof(router_enable)));

// ESP_NETIF_SUBNET_MASK, DHCP Option 1, Configure the subnet mask
// If it fails to configure the subnet mask via ESP_NETIF_SUBNET_MASK,␣
↪please make modifications using esp_netif_set_ip_info
```

[Performance] How to test the bit rate of Wi-Fi modules?

Please use the codes in example `example/wifi/iperf` provided by ESP-IDF SDK.

[LWIP] 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?

- The default network segment used by ESP8266 SoftAP is 192.168.4.*, and its IP address is 192.168.4.1. When connecting ESP8266 to the router of 192.168.4.X, it cannot distinguish whether this address indicates its own SoftAP or the external router.

[Connect] How many devices is ESP8266 able to connect in SoftAP mode?

Up to eight devices in hardware level. However, to ensure module performance, it is recommended to connect four devices at most.

Do ESP8266/ESP32/ESP32-S2 support web/SoftAP provisioning?

Yes.

- For ESP8266, please refer to example ESP8266 softap_prov.
- For ESP32/ESP32-S2, please refer to example ESP32/ESP32-S2 softap_prov.

[Connect] How do ESP8266 and ESP32 hide SSID in SoftAP mode?

The variable ssid_hidden in wifi_ap_config_t structure can be configured to hide the SSID.

Does the buffer parameter in esp_wifi_802.11_tx interface include FCS?

No, the FCS frame is generated automatically by hardware.

What is the supported Wi-Fi frequency band and power meter for ESP-WROOM-32D?

The Wi-Fi frequency band is 2412 ~ 2484 MHz, and the available channels and corresponding operating frequencies can be configured in software. There are default values in power meter, and it can also be configured by software. For detailed guidance, please refer to ESP32 Phy Init Bin Parameter Configuration Guide.

What is the maximum value of ESP32 Wi-Fi RF power

The RF power of ESP32 is 20 dB, which is exactly the maximum value.

How does ESP32 adjust Wi-Fi TX power?

- Configure Component config -> PHY -> Max Wi-Fi TX power(dBm) via menuconfig, and the max value is 20 dB.
- Use API *esp_err_t esp_wifi_set_max_tx_power(int8_t power);*.

[Connect] How many devices is ESP32 able to connect in AP mode?

Up to 10 devices in AP mode. It is configured to support four devices by default.

[Connect] How do Wi-Fi modules rank signal strength levels based on RSSI values

We do not have a rating for RSSI signal strength. You can take the calculation method from Android system for reference if you need a standard for classification.

```
@UnsupportedAppUsage
private static final int MIN_RSSI = -100;

/** Anything better than or equal to this will show the max bars. */
@UnsupportedAppUsage
private static final int MAX_RSSI = -55;

public static int calculateSignalLevel(int rssi, int numLevels) {
  if(rssi <= MIN_RSSI) {
    return 0;
  } else if (rssi >= MAX_RSSI) {
    return numLevels - 1;
  } else {
    float inputRange = (MAX_RSSI -MIN_RSSI);
    float outputRange = (numLevels - 1);
    return (int)((float)(rssi - MIN_RSSI) * outputRange / inputRange);
  }
}
```

[Connect] Why does ESP32 disconnect from STA when it is in Soft-AP mode?

- By default, the ESP32 will disconnect from the connected STA if it doesn't receive any data from this STA for continuous 5 minutes. This time can be modified via API esp_wifi_set_inactive_time.
- Note: esp_wifi_set_inactive_time is a newly added API.
  - master commit: `63b566eb27da187c13f9b6ef707ab3315da24c9d`
  - 4.2 commit: `d0dae5426380f771b0e192d8ccb051ce5308485e`
  - 4.1 commit: `445635fe45b7205497ad81289c5a808156a43539`
  - 4.0 commit: Pending, the MR is not merged yet
  - 3.3 commit: `908938bc3cd917edec2ed37a709a153182d511da`

[Connect] While ESP32 connecting Wi-Fi, how can I determine the reason of failure by error codes?

For ESP-IDF v4.0 and later versions, please refer to the following codes to get the reason

```
if (event_base == WIFI_EVENT && event_id == WIFI_EVENT_STA_DISCONNECTED) {
  wifi_event_sta_disconnected_t *sta_disconnect_evt = (wifi_event_sta_
↪disconnected_t*)event_data;
  ESP_LOGI(TAG, "wifi disconnect reason:%d", sta_disconnect_evt->reason);
  esp_wifi_connect();
  xEventGroupClearBits(s_wifi_event_group, CONNECTED_BIT);
}
```

When the callback function received `WIFI_EVENT_STA_DISCONNECTED` event, you can get the reason through the `reason` variable from wifi_event_sta_disconnected_t.

- `WIFI_REASON_AUTH_EXPIRE`: This code is returned during the auth phase when the STA sends an auth but do not received any auth reply from the AP within the specified time. The possibility of this code occurrence is low.
- `WIFI_REASON_AUTH_LEAVE`: This code is sent by AP, normally because the AP is disconnected from the STA for some reason.
- `WIFI_REASON_4WAY_HANDSHAKE_TIMEOUT` or `WIFI_REASON_HANDSHAKE_TIMEOUT`: Wrong password.

  `WIFI_REASON_4WAY_HANDSHAKE_TIMEOUT` is the standard generalized error code, while `WIFI_REASON_HANDSHAKE_TIMEOUT` is a customized error code. The main difference is: `WIFI_REASON_4WAY_HANDSHAKE_TIMEOUT` occurs when the router tells the device the password is wrong; `WIFI_REASON_HANDSHAKE_TIMEOUT` occurs when the device itself performs a timeout mechanism without being informed about the wrong password by the router.
- `WIFI_REASON_CONNECTION_FAIL`: This code is returned during the scan phase when the STA scanned a matched AP while the AP is in the blacklist. This is probably because that the AP has actively disconnected from the STA last time or something wrong happened when the STA connecting the AP.

Does ESP32 perform domain name resolution each time it connects to the server?

The domain name is resolved via DNS within the stack, and the resolved data will be cached within the specified time. The cache time is based on the TTL data obtained from the DNS server, which is a parameter filled when configuring the domain name, usually 10 minutes.

[Connect] What does the number after the state machine switch in Wi-Fi log mean?

eg: run -> init (fc0), fc0 means the STA has received the deauth frame and reason is password error.

- c0 indicates the received frame type (00 indicates a timeout)
- f indicates reason

Frame type: [a0 disassoc], [b0 auth], [c0 deauth].

[Connect] What does bcn_timeout, ap_probe_send_start mean

The STA does not receive the Beacon frame within the specified time (6 s by default for ESP32, equals to 60 Beacon Intervals).

- The reason could be:
  - Insufficient memory. "ESP32_WIFI_MGMT_SBUF_NUM" is not enough (there will be errors like "esf_buf: t=8, l=beacon_len, . . ." in the log). You can check this by typing the heap size when received a Disconnect event.
  - The AP did not send a beacon. This can be checked by capturing beacons from AP.
  - Rssi too low. When the Rssi value is too low in complex environments, the STA may not receive the beacon. This can be checked by retrieving Rssi values via `esp_wifi_sta_get_ap_info`.
  - Hardware related issues. Bad package capturing performance.

When there is a bcn_timeout, the STA will try to send Probe Request for five times. If a Probe Response is received from the AP, the connection will be kept, otherwise, the STA will send a Disconnect event and the connection will fail.

[Connect] How to reconnect Wi-Fi after it disconnected?

Call esp_wifi_connect after received the `WIFI_EVENT_STA_DISCONNECTED` event.

[Connect] When does ESP32 disconnect from SoftAP in station mode

By default, the ESP32 will disconnect from the AP if it does not receive any beacon for 6 s. This time can be modified via esp_wifi_set_inactive_time.

[Scan] Why does the STA cannot find any AP sometimes during the scanning?

Generally, it is because the AP is too far away from the STA. Sometimes this can also be caused by inappropriate configurations of the scanning parameters.

[Scan] What is the maximum number of APs that can be scanned

There is no limit to the maximum number of APs that can be scanned. The number depends on how many APs are around and configurations of the scanning parameters, such as the time spent on each channel, the longer time spent on each channel the more likely it is to find all the APs.

[Scan] Can I choose to connect the best AP when there are multiple APs with identical ssid/password during the scan

By default, the scan type is WIFI_FAST_SCAN, which makes the STA always connects the first AP during the scan. If you expect to connect the best AP, please set scan_method to WIFI_ALL_CHANNEL_SCAN and configure sort_method to determine whether to choose the AP with the strongest RSSI or connect to the most secure AP.

[Scan] How to configure scan_method in the wifi_sta_config_t structure? What is the difference between all_channel_scan and fast_scan?

all_channel_scan and fast_scan are used to find the appropriate AP before connecting. The scan_method is set to fast_scan by default, which is mainly used together with threshold to filter APs with weak signal or encryption.

- When fast_scan is set, the STA will stop scanning once it finds the first matched AP and then connect to it, so as to save time for connection.
- When all_channel_scan is set, the STA will scan all channels and store four APs with the best signal or the most secure encryption according to the sorting method configured in sort_method. After the scan is completed, the STA will connect one of the four APs with the best signal or the most secure encryption.

[LWIP] How to get error code of the socket?

- For ESP-IDF v4.0 and later versions: use the value of *errno* directly to get the error code after the socket API returns failure.
- For previous versions of ESP-IDF v4.0: call *getsockopt(sockfd, SOL_SOCKET, SO_ERROR, . . .)* to get the error code after the socket API returns failure, otherwise you may get wrong error code when multiple sockets operate simultaneously.

[LWIP] What is the default keep-alive time of TCP?

By default, a TCP keep-alive message will be sent every 75 seconds for 9 times if no TCP message is received for two consecutive hours. Then, if there is still no message received, the LWIP will disconnect from the TCP.

The keep-alive time can be configured via socket option.

[LWIP] What is the retransmission interval of TCP

When ESP32 serves as the transmitter, the first retransmission interval is normally 2 3 s by default. Then, the next interval is determined by Jacoboson's algorithm, which can be simply seen as a multiplication of 2.

[LWIP] What is the maximum number of sockets that can be created?

32 for most, and the default number is 10.

[Sleep] What kinds of sleeping mode does ESP32 have? What are the differences?

- There are mainly three sleeping modes: Modem sleep, Light sleep and Deep sleep.
  - Modem sleep: the station WMM sleeping mode specified in the Wi-Fi protocol (the station sends NULL data frame to tell the AP to sleep or wake up). The Modem sleep mode is enabled automatically after the station connected to AP. After entering this mode, the RF block is disabled and the station stays connected with the AP. The Modem sleep mode will be disabled after the station disconnected from the AP. The ESP32 can also be configured to decrease the CPU's clock frequency after entering Modem sleep mode to further reduce its current.
  - Light sleep: this is a station sleep mode based on Modem sleep mode. The difference between is that, besides for the RF block being disabled, the CPU will also be suspended in this mode. After exiting from Light sleep mode, the CPU continues to operate from where it stopped.
  - Deep sleep: a sleeping mode un-specified in the Wi-Fi protocol. During Deep sleep mode, all the blocks except for RTC is disabled, and the station cannot be connected to AP. After exiting from this mode, the whole system will restart to operate (similar to system restart).

[Sleep] Where to enable the speedstep function for ESP32 in modem sleep mode?

Go to menuconfig -> Component Config -> Power Management.

[Sleep] How low can the speedstep function go for ESP32 in modem sleep mode

For now, the CPU clock can go down to as low as 40 MHz.

[Sleep] What affects the average current of ESP32 in modem sleep mode?

The main factors are: the core, the clock frequency and the percentage of idle time of the CPU, whether there is Wi-Fi data sent or received during the test, data sending or receiving frequency, the transmitting power of RF block, whether the time when the router sends beacon is accurate, whether there are peripheral modules working, and etc.

[Sleep] Why the average current measured in modem sleep mode is a bit high?

- A lot of Wi-Fi data sent and received during the test. The more data there is, the less chance there will be for entering sleeping mode and the higher average current will be.
- The time when the router sends out beacon is not accurate. The station needs to wake up and monitor the beacon regularly, thus it will wait longer if the beacon time is not accurate. In this way, the station has less time in sleeping mode and the average current will be high.
- There are peripheral modules working during the test. Please close them before the test.
- The station+SoftAP mode is enabled. During modem sleep state, the current will only be lower in station-only mode.

[Sleep] Why the average current measured in light sleep mode is a bit high?

Besides for the reasons listed in the last question, the possible reasons also could be:

- The application layer code is running continuously, thus the CPU does not get chance to suspend.
- The application layer has enabled ets timer or esp timer and the timeout interval is short, thus the CPU does not get chance to suspend.

Does ESP8266 support 802.11k/v/r protocol?

For now, the ESP8266 only supports 802.11k and 802.11v, please refer to example roaming.

[Connect] After the NONOS_SDK updated from version *2.1.0* to *2.2.2*, why does the connecting time become longer

Please update to version *master*, which has solved the incompatibility issue between the CCMP encryption and some APs.

How does ESP32 receive and transmit Wi-Fi 802.11 packets?

- By using the following APIs:

```
esp_err_t esp_wifi_80211_tx(wifi_interface_t ifx, const void *buffer, int␣
↪len, bool en_sys_seq);
esp_wifi_set_promiscuous_rx_cb(wifi_sniffer_cb);
```

- The abovementioned APIs are also used in the ESP-MDF project, please refer to mconfig_chain.

[Connect] The ESP32 and ESP8266 failed to connect to router, what could be the reasons

- Please check if the SSID or password is wrong.
- There could be errors in different Chinese codes, so it is not recommended to use an SSID written in Chinese.
- The settings of bssid_set. If the MAC address of the router does not need to be identified, the stationConf.bssid_set should be configured to 0.
- It is recommended to define the wifi_config field in wifi_config_t using the static variable *static*.

[Connect] What kind of networking methods does ESP8266 have

- SmartConfig mode: using SmartConfig. The device scans feature pack in sniffer mode.
- SoftAP mode: the device enables SoftAP and sends SSID and password after the phone connects to SoftAP and set up a stable TCP/UDP connection.
- WPS mode: an additional button should be added on the device; or using the phone to enable WPS after it connected to SoftAP.

[Connect] What are the specifications of Wi-Fi parameters when using SmartConfig?

According to *wifi spec*, the SSID should not exceed 32 bytes and its password should not exceed 64 bytes.

[Connect] Does ESP8266 Wi-Fi support WAP2 enterprise-level encryption

- Yes. Please refer to example wpa2_enterprise.
- To build RADIUS server, you can use FreeRADIUS.

[Connect] What are the low-power modes for ESP32 to maintain its connection to Wi-Fi?

- In such scenarios, the chip switches between Active mode and Modem-sleep mode automatically, making the power consumption also varies in these two modes.
- The ESP32 supports Wi-Fi keep-alive in Light-sleep mode, and the auto wakeup interval is determined by the DTIM parameter.
- Please find examples in ESP-IDF - > examples - > wifi - > power_save.

Do Espressif's chips support WPA3?

- ESP32 series: WPA3 is supported from esp-idf release/v4.1 and enabled by default. Go to menuconfig > Component config > Wi-Fi for configuration.
- ESP8266: WPA3 is supported from the master branch of ESP8266_RTOS_SDK and enabled by default. Go to menuconfig > Component config > Wi-Fi for configuration.

[Connect] How does the device choose AP when there are multiple identical SSIDs in the current environment?

- The device connects to the first scanned AP.
- If you expect to sort APs by signal quality and etc., use the scan function to filter manually.
- If you expect to connect to a specified AP, add BSSID information in connection parameters.

[Connect] Does ESP8266 have repeater solutions?

- We have not officially released such application solutions yet.
- For relay related applications, please find on github. The relay rates should be set basing on real tests.

What is ESP-NOW? What are its advantages and application scenarios?

- ESP-NOW is a kind of connectionless Wi-Fi communication protocol that is defined by Espressif.
- In ESP-NOW, application data is encapsulated in action frames from different vendors and then transmitted from one Wi-Fi device to another without a connection.
- ESP-NOW is ideal for smart lights, remote control devices, sensors and other applications.

What is the retransmission time for ESP32's data frame and management frameCan this be configured

The retransmission time is 31 and it can not be configured.

How to customize the hostname for ESP32

- Taking ESP-IDF V4.2 as an example, you can go to menuconfig > Component Config > LWIP > Local netif hostname, and type in the customized hostname.
- There may be a slight difference on naming in different versions.

How to obtain 802.11 Wi-Fi packets

Please refer to Wireshark User Guide in ESP-IDF Programming Guide.

Does ESP32 Wi-Fi support PMF (Protected Management Frames) and PFS (Perfect Forward Secrecy)

The PMF is supported both in WPA2 and WPA3, and PFS is supported in WPA3.

How to get the RSSI of the connected AP for ESP32 IDF v4.1 Wi-Fi?

- It can be obtained via scanning, please refer to example scan.
- If there are multiple identical SSIDs in the current environment, you can get the AP's bssid first after it connects to an AP and specify the bssid via wifi_scan_config_t, then get the RSSI by calling esp_wifi_scan_start().

  Code:

```
//Obtain bssid via WIFI_EVENT_STA_CONNECTED in the callback function event_handler()
else if(event_base == WIFI_EVENT && event_id == WIFI_EVENT_STA_CONNECTED) {

        wifi_event_sta_connected_t* sta_connected_event = (wifi_event_sta_connected_t*) event_data;
        ESP_LOGI(TAG, "AP MAC:"MACSTR"", MAC2STR(sta_connected_event->bssid));
        ...
        //specify a bssid to perform scanning
        wifi_scan_config_t wifi_scan_config = {
            .bssid = sta_connected_event->bssid,
        };
        ESP_ERROR_CHECK(esp_wifi_scan_start(&wifi_scan_config, true));
        ...
}
```

Why does ESP8266 print out an AES PN error log when using esptouch v2?

This occurs when ESP8266 has received retransmitted packets from the router for multiple times. However, this will not affect your usage.

Hardware related

[]

Chip comparison

[]

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.

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.

Development board

[]

Hardware design

[]

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.

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.

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.

RF related

[]

Process and ESD

[]

Production test

[]

Espressif Q & A.

Test verification

[]

Power consumption verification

[]

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.

Commercial FAQ

[]

Which certificates have your products passed?

Please check our Certificates , where you can get all the relevant information about our products.

Does your company have the ISO Quality Management System Certification?

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

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.

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.

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: sales@espressif.com.

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 .

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 .

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.

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.

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.

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 .

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.

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 .

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.

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.

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.

1.2 Document contribution

[]

We welcome all the contributions to esp-faq project, such as fixing bugs, adding new documents and etc. We will accept new requests via Github Pull Requests.

1.2.1 Commit process

This section provides a brief overview of the `Add new items` and `Modify contents` processes. For the specific requirements during the processes, please refer to the links provided.

Here, we do not provide further operational instructions on `git`, please see Git learning material.

Add new items

1. *Create a new branch* following the *Branch naming conventions*;
2. Find the corresponding `*.rst` file locally or on web IDE, then add new items according to the template formats;
3. After your writing finished, you can check the document in the preview interface and build it using *Local build environment* to see if it has any formatting issues;
4. Push your branch to github and commit a pull request following the *Commit message standards*;
5. If all the abovementioned steps are finished following requirements, then you can *Submit a merge request*;
6. After all the review comments resolved and new pull requests updated, then this process is fully completed.

Modify contents

1. *Create a new branch* locally following the *Branch naming conventions*;
2. Find the corresponding `*.rst` file locally or on web IDE, then modify the desired contents;
3. After the modification finished, you can check the document in the preview interface and build it using the *Local build environment* to see if it has any formatting issues;
4. Push your branch to github and commit a pull request following the *Commit message standards*;
5. If all the abovementioned steps are finished following requirements, then you can *Submit a merge request*;
6. After all the review comments resolved and new pull requests updated, this process is fully completed.

1.2.2 Create a new branch

All the new branches are based on the **master branch**, so please make sure your current branch is the one you expect to merge.

For example:

```
git status #To see the status of your current branch
git checkout -b add/artificial-intelligence_camera_model #To add new contents about
↪"artificial-intelligence camera model"
```

1.2.3 Branch naming conventions

- Add a new item: `add/artificial-intelligence_{q&a}`, `{q&a}` is the brief English name of the file. For example, if you expect to add a new item as `artificial intelligence camera model`, then the branch name should be: `add/artificial-intelligence_camera_model`.
- Modify contents: `mod/artificial-intelligence_q&a`, `q&a` is the brief English name of the file. For example, if you expect to modify the contents about `artificial intelligence camera model`, then the branch name should be: `mod/artificial-intelligence_camera_model`.

1.2.4 Q&A template

Please do updates according to the template formats as follows.

Q&A code example

```
--------------

Does Espressif have any AI image recognition products?
--------------------------------------------------------

Yes, we already have the ESP-EYE development board. With ESP32 as its main␣
↪control chip, ESP-EYE supports various types of cameras, such as 0v2640,␣
↪3660, 5640 and etc.
```

Q&A figure example

```
--------------

curses.h: No such file or directory
-------------------------------------------

Screenshot: support ESP8266 chip, but ESP8266_RT

.. figure:: _static/application-solution/android-application/case_two_
↪kconfig_error.png
    :align: center
    :width: 900
    :height: 100

Solution: sudo apt-get install libncurses5-dev
```

1.2.5 Local build environment

- Use ubuntu or Debian system as test environment, and configure your python version to `3.7`.
- It is recommended to use python virtual environment or docker environment.

```
# Install python3.7 and virtual environment

sudo apt-get install python3.7 python3.7-venv

# Create virtual environment

python3.7 -m venv ~/.pyenv3_7

# Activate virtual environment

source ~/.pyenv3_7/bin/activate

# Upgrade pip

pip install --upgrade pip

# Install pip component

pip install -r docs/requirements.txt

# build the Chinese version

cd docs/cn/ && make html && cd -

# Build the English version

cd docs/en/ && make html && cd -

# Exit virtual environment

deactivate
```

1.2.6 Commit message standards

Please add commit messages on your branch to explain what you have added/modified/deleted. Each commit has one message, for example:

```
artificial-intelligence: add esp-eye support those camera models

1. esp-eye support those camera models.
```

The first line of the commit message should be like "Q&A category: add/fix/modify/delete something". And this line should be started with the file name you updated, for example:

`artificial-intelligence:  esp-eye support those camera models.`

If more information should be added into the commit message, please add it in the later commits after the first line.

A good commit message should tell why this update came up, thus making others get to know about this project when reading these commit logs. It may seem like a waste of time to write a good commit message, but it will be useful for you when trying to know why something changed.

1.2.7 Submit a merge request

Once your updates finished, you can conduct the first commit of your branch. Please add more commits if you need to do further updates. After finishing all the commits on this branch, you are ready to submit a merge request.

We use the github "Merge Requests" function to merge your branch into the master, the steps include:

1. Push your branch to the github repository;
2. Go to esp-faq and click "New pull request";
3. Select the branch that you created and waited for merge, and fill detailed information in the "Merge Request".

Please see IDF Contribution Guide.

Merge request specifications

- Title:

```
add: a brief overview
```

- Description:

  Describe the updates of this merge request in points.

- For example

Title:

```
artificial-intelligence: add esp-eye support those camera models.
```

Description:

```
1. add esp-eye support those camera models.
```