CONTENTS

1 Instruction
  1.1 Question search .................................................. 3
  1.2 Document contribution ............................................ 129
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.

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Development environment</th>
<th>Application solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software framework</td>
<td>Hardware related</td>
<td>Test verification</td>
</tr>
</tbody>
</table>
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.

When using the Arduino IDE development platform, how to read the MAC address of the Wi-Fi that comes with ESP32?

- Please refer to the “esp32-arduino development framework”.
- Use “WiFi.macAddress()” to obtain the MAC address of ESP32’s Wi-Fi.
- Please refer to example: Serial.println(WiFi.macAddress());.

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:

<table>
<thead>
<tr>
<th>Pins</th>
<th>GPIO12</th>
<th>GPIO0</th>
<th>GPIO2</th>
<th>GPIO4</th>
<th>GPIO15</th>
<th>GPIO5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

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_world.elf, the instruction should be as follows:
**How to do RF performance test with ESP32&ESP8266&ESP32S2?**

- Please refer to ESP RF Test Guide.

**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 panicked (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 \-0xffffffff). 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`

I found “Brownout detector was triggered” failure on my ESP32. How to resolve such issue?

- The ESP32 has a built-in brownout detector which can detect if the voltage is lower than a specific value. If this happens, it will reset the chip in order to prevent unintended behaviour.
- This message may be reported in various scenarios, while the root cause is that the chip with a power supply has momentarily or permanently dropped below the brownout threshold. Please try replacing power supply, USB cable, or installing capacitor on power supply terminals of your module.
- You can do configuration to reset the threshold value or disable the brownout detector. Please refer to `config-esp32-brownout-det` for details.

After imported the `protocol_examples_common.h` header file, how come it cannot be found while compiling?

**CHIP: ESP32**

- Please add `set(EXTRA_COMPONENT_DIRS $ENV{IDF_PATH}/examples/common_components/protocol_examples_common)` in CMakeLists.txt under the project.

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 occurred”?

Please check your CMake version first using cmake --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, errors 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
```

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

Do I need to use command idf.py set-target every time I switch to another project?

When building the project with idf.py build, the target is determined as follows:

1. If the build directory already exists, we will use the target the project was previously built for. It is stored in CMakeCache.txt file in the build directory.
2. Alternatively, if the build directory doesn’t exist, we will check if the sdkconfig file exists, and use the target specified there.
3. If both the build directory and sdkconfig file exists, and specify different targets, we will report an error. This shouldn’t happen normally, unless sdkconfig was changed manually without deleting the build directory.
4. If neither sdkconfig file nor build directory exists, we will consider IDF_TARGET variable, which can be set either as CMake variable or as an environment variable. If this variable is set and is different from the target specified in sdkconfig or in the build directory, we will also report an error.
5. Finally, if sdkconfig doesn’t exist, build directory doesn’t exist, and the target is not set via IDF_TARGET, then we will use the default value. The default value can be set in sdkconfig defaults.
6. If it isn’t set using any of the above methods, then we will build for esp32 target.

To answer your question:

- Once the project is configured and built once for a certain target, it’s not necessary to run `idf.py set-target` again other than to switch to a different target. `idf.py set-target` stores the selected target in the project’s build directory and `sdkconfig` file, not in the terminal environment. So if you switch to a different directory and build another project, then come back, the target will not change, and will be the same as previously set for this project.

- If you want to make the project built for certain target by default, add `CONFIG_IDF_TARGET="esp32s2"` to the `sdkconfig.defaults` file of the project. After this, if `sdkconfig` file doesn’t exist and build directory doesn’t exist, `idf.py` build command will build for that target specified in `sdkconfig.defaults`.

- `idf.py set-target` command can still be used to override the default target set in `sdkconfig.defaults`.

---

**How to know the version of ESP-IDF, is it recorded in a certain document?**

- There is a variable `IDF_VER`, you can call the function `esp_get_idf_version` to check.

- Please refer to “components/esp_common/include/esp_idf_version.h” to see more details.

---

**How to optimize ESP-IDF compilation in windows environment?**

**CHIP: ESP32**

- Please add the directories of ESP-IDF source code and compiler `.espressif` to the exclusions of anti-virus program.

---

**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:

<table>
<thead>
<tr>
<th>Modules</th>
<th>3V3</th>
<th>GND</th>
<th>TXD</th>
<th>RXD</th>
<th>IO0</th>
<th>EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial tool</td>
<td>3V3</td>
<td>GND</td>
<td>RXD</td>
<td>TXD</td>
<td>DTR</td>
<td>RTS</td>
</tr>
</tbody>
</table>

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?

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

How to program ESP32-C3’s firmware offline?

- Download the latest Flash Download Tools from espressif.com. Versions after v3.8.8 and later versions already support ESP32-C3 series programming.

How does ESP32 set Flash SPI to QIO mode?

- It can be set in configuration terminal through “menuconfig -> Serial flasher config -> Flash SPI mode”, the corresponding API is esp_image_spi_mode_t().

After downloaded program and powered on EPS8266, the serial port prints the following log. What is the reason?

```plaintext
ets Jan 8 2013, rst cause:1, boot mode:(7,7)
waiting for host
```

- `waiting for host` means the Boot is in SDIO mode, indicating that GPIO15 (MTDO) is pulled up (HIGH), please refer to ESP8266 Boot Mode Description.

When using UART to upgrade firmware for ESP32, how to set two `app_main` spaces?

- Sorry, ESP32 chips do not support setting two `app_main` spaces when upgrading firmware via UART.
- The mechanism of UART upgrading firmware is the same as that of flash download tool. The upgraded firmware will directly replace the old one, and it is not supported to keep the old firmware.

What are the Espressif module programming tools?

- For Espressif programming software, you can go to this webpage and download it: flash download tool. Only Windows environment can support no GUI tool embedded.
- Espressif programming tool esptool is written based on python and open source code, supports secondary development.
What is the difference between the Factory and Developer modes of the flash download tool?

- Factory mode supports multi-channel downloads, while Developer mode only supports single channel.
- The path of bin files under Factory mode is relative, while under Developer is absolute.

Why does the programming failed for the jig with a 4-port hub in factory mode?

:CHIP: ESP32 | ESP8266:
- It is because Espressif products complete the calibration operation through transmitting some packets when starting up. This operation requires 3.3 V voltage and a guaranteed peak current of 500 mA. Therefore, when it comes to more than one ports, there will be situations where the computer cannot program or the programming is interrupted due to the insufficient power supply of the computer’s USB when programming via connecting to a computer’s USB. It is recommended to use the hub for programming and supply power to the hub in the meantime.

I’m using an ESP32-WROVER-B module to download the AT firmware via the flash download tool. However, an error occurred after writing to flash. But the same operation succeeded when replacing the module with ESP32-WEOVER-E, what is the reason?

- The ESP32-WROVER-B module leads out the SPI Flash pin, but the ESP32-WROVER-E module does not. Please check whether the SPI Flash pin of the ESP32-WROVER-B module is re-used by other external application circuits.
- Connecting the CMD pin of the SPI Flash in ESP32-WROVER-B to GND will cause the flash failing to start. And the following error log will be printed:

```
rst:0x10 (RTCWDT_RTC_RESET),boot:0x1b (SPI_FAST_FLASH_BOOT)
flash read err, 1000
ets_main.c 371
ets Jun 8 2016 00:22:57
```

The encrypted device cannot be re-flashed via the flash download tool, what is the reason?

:CHIP: ESP32 | ESP32-S2:
- Currently, an encrypted device cannot be flashed again using the flash download tool. It only supports one-time encryption of plaintext.
When updating ESP32 firmware through UART interface based on esptool serial port protocol, can I add a new app partition?

- The partitions in flash depend on the data in partition_table.bin. If partition_table.bin can be updated, the storage space of other data, such as bootloader.bin and app.bin, can be redivided to create an app partition.

**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 OV2640, 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.

**Does ESP modules support TensorFlow?**

**CHIP: ESP32 | ESP32-S3**

- Yes, please refer to Github tensorflow examples and blog for instructions.

**AT**

[]

**How to test and optimize the throughput of ESP32 AT?**

- Many factors are affecting the AT throughput test. It is recommended to use the iperf example in esp-idf for testing. While testing, please use the passthrough mode, adjust the data length to 1460 bytes, and send data continuously.

- If the test rate does not meet your requirements, you can increase it by modifying the menuconfig iperf parameter in esp-idf and compiling your esp-at project.
How long does it take for the ESP32 AT to connect to Wi-Fi?

How long does it take for ESP32 AT (firmware release/v2.0.0.0) to connect to Wi-Fi after firmware initialization, or module startup?

In an office scenario, the connection time is 11 s. However, in actual practice, Wi-Fi connection time depends on the router performance, network environment, module antenna performance, etc.

Why does AT prompt “busy”?

- The “busy” prompt indicates that the previous command is being executed, 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:

- If you enter AT+GMR (line break CR LF) + (space) through serial ports, the system will execute AT+GMR (line break CR LF) immediately, because the it is already considered to be a complete AT command. 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 the space is an incorrect command.

Is it possible to change the TCP send window size in AT firmware?

- Currently, it cannot be changed by commands, but you can compile the esp-at code to generate a new firmware.

- You can reconfigure the menuconfig parameter: Component config -> LWIP -> TCP -> Default send buffer size.

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


- You can develop more AT commands based on the Espressif’s official esp-at project.
**Why is there a “no module named yaml” error when I am compiling the ESP32 AT?**

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

**Does the default AT firmware for ESP32 modules support Bluetooth functionality?**

- For the ESP32-WROOM series of modules, the default AT firmware does not support Classic Bluetooth AT commands, but you can enable this functionality in menuconfig (Component config -> AT -> [*] AT bt command support) and compile your firmware based on the esp-at project.
- For the ESP32-WROVER series of modules, the default AT firmware support Classic Bluetooth AT commands.

**Does the ESP8266 AT support PSK authentication in SSL connections?**

Support in IDF-based AT, but not in ESP8266_NONOS_SDK-based AT.

- AT firmware.
- ESP8266_NONOS_SDK-based AT: ESP8266 AT Instruction Set.
- IDF-based AT: RTOS AT Command Set.

**Can the serial port baudrate be modified in AT Commands? (Default: 115200)**

Yes, you can use either of the two ways below to modify it:

- Use the command `AT+UART_CUR` or `AT+UART_DEF`. See AT Instructions for more information.
- Re-compile the AT firmware: establish the compiling environment and change the UART baudrate.

**How does ESP8266 establish an SSL connection using AT commands?**

- Please refer to the following commands to establish an SSL connection between ESP8266 and a server:

  ```
  AT+CWMODE=1 // set Wi-Fi mode to station
  AT+CWJAP="espressif_2.4G","espressif" // connect to AP, enter ssid and password
  AT+CIPMUX=0 // enable the single connection
  AT+CIPSTART="SSL","www.baidu.com",443 // establish an SSL connection
  ```
How do I specify the TLS protocol version for ESP32 AT?

When compiling the esp-at project, you can disable the unwanted versions in the menuconfig -> Component config -> mbedTLS.

What interfaces of ESP chips can be used to transmit AT commands?

ESP8266, ESP32, ESP32-S2 can transmit AT commands through SDIO, SPI, and UART. You can configure it in menuconfig -> Component config -> AT when compiling the esp-at project.

How does the ESP32 AT communicate through the UART0 port?

The default AT firmware communicates through the UART1 port. If you want to communicate through UART0, please download and compile the esp-at project.

- Refer to How to clone and compile a project to set up the compiling environment;
- Modify the module’s UART pins in your factory_param_data.csv, i.e. change uart_tx_pin to GPIO1 and uart_tx_pin to GPIO3;
- Configure your esp-at project: make menuconfig > Component config > Common ESP-related > UART for console output(Custom) > Uart peripheral to use for console output(0-1)(UART1) > (1)UART TX on GPIO# (NEW) > (3)UART TX on GPIO# (NEW).

How to wake up ESP8266 from Light-sleep mode using AT commands?

Please refer to the command AT+SLEEPWKCFG.

How to use AT commands to establish the Bluetooth LE passthrough between ESP32-SOLO-1C and a cell phone?

- Configure ESP32-SOLO-1C as the Bluetooth LE passthrough server. See Bluetooth LE AT Examples for detailed steps.
- Download a Bluetooth LE test tool in your cell phone, such as nRF Connect (Android) and lightblue (iOS), then open SCAN to find the MAC address of the ESP device, and finally you can send commands.
My ESP8266 uses the compiled ESP-AT firmware and needs the OTA function. How much flash size is needed?

The ESP-AT firmware needs at least 2 MB (16 Mbit) of flash if OTA function is required, and at least 1 MB (8 Mbit) of flash if the OTA function is not required.

How to enable the MDNS function using AT commands?

Use the AT+MDNS command to enable the MDNS function.

| AT+CWMODE=1                     // set the device to station mode |
| AT+CWJAP="ssid","password"     // connect to an AP, enter ssid and password |
| AT+MDNS=1,"esp"," *pos.* tcp.",3030 // enable MDNS |
| AT+MDNS=0                      // disable MDNS |

Does esp-at firmware support MQTT?

- ESP8266 firmware supports MQTT in v2.1.0.0-rc1 and later versions.
- ESP32 firmware supports MQTT in v2.0.0.0 and later versions.
- See release notes for more details.

After MCU sends AT+CIPSEND, it receives the busy p... response. Does the MCU need to resend the data?

busy p.. means the previous command is being executed and the current input is invalid. It is recommended to wait for the response of the previous command before resending AT+CIPSEND.

Does ESP-WIFI-MESH support AT commands?

Currently ESP-WIFI-MESH does not support AT commands.

How does ESP32 enable the BluFi functionality in AT?

- The default AT firmware does not support the BluFi function, but you can enable it by compiling the esp-at code to generate a new firmware.
- When compiling, enable the BluFi functionality in menuconfig: make menuconfig—>Component config—>AT—>[*] AT blufi command support.
How to get the default firmware version in modules or development boards?

- The factory firmware varies from modules to modules, or boards to boards. You can ask your purchaser to consult Espressif’s Business Support.
- If the module is shipped with AT firmware, you can use the command AT+GMR to check the version information.

Are there any examples of using AT commands to connect to aliyun or Tencent Cloud?

- Download and flash AT firmware.
- Aliyun: AT+MQTT aliyun.
- Tencent Cloud: AT+MQTT cloud.

Does AT firmware support SSL certificate authentication?

- Yes, please refer to SSL certification commands for more information.
- For how to generate the SSL certificate bin, please refer to esp-at/tools/README.md.
- The flash address of the SSL certificate is determined by at_customize.csv.

Does AT support websocket commands?

- Not supported in the default firmware.
- It can be implemented by custom commands. See websocket and How_to_add_user-defined_AT_commands for more information.

Does the AT firmware shipped in modules support flow control?

- Hardware flow control is supported, but software flow control not.
How to modify the number of TCP connections in AT?

- The ESP32 AT supports a maximum of 16 TCP connections, which can be configured in menuconfig as follows:
  - make menuconfig—> Component config—> AT—> (16)AT socket maximum connection number
  - make menuconfig—> LWIP—> (16)Max number of open sockets
- The ESP8266 AT supports a maximum of 5 TCP connections, which can be configured in menuconfig as follows:
  - make menuconfig—> Component config—> AT—> (5)AT socket maximum connection number
  - make menuconfig—> LWIP—> (10)Max number of open sockets

How to view the error log of AT firmware?

- For ESP32, the error log is output through the download port. By default, UART0 is GPIO1 and GPIO3.
- For ESP8266, it is output from UART1 TX, which is GPIO2 by default.
- See AT Hardware Connection for more details.

How to OTA upgrade AT firmware?

- You can use the following AT commands to do it:

```
AT+CNM=1
AT+CNJAP_DEF="ssid","password"
AT+CIUPDATE
```

How does the ESP32 module use AT commands to implement encrypted Bluetooth pairing?

- The commands to encrypt Bluetooth pairing are as follows:

```
AT+RST // restart the module
AT+GMR // check the module version information
AT+BLEINIT=2 // initialize the module as a server
AT+BLEGATTSSRVCRE // GATTS creates services
AT+BLEGATTSSRVSTART // GATTS start services
AT+BLEADDR? // query the public address of the Bluetooth LE device
AT+BLEADV=50,50,0,0,4 // set Bluetooth LE advertising parameters
AT+BLEADVDATA="020120" // set Bluetooth LE advertising data
AT+BLESEC=4,1,8,3,3 // set encrypted parameters
AT+BLEADVSTART // start the Bluetooth LE advertising
AT+BLEDV=0,3 // After connecting without a secret key, use this command to generate an encrypted connection request and an encryption key
```
ESP-FAQ

What is the default Wi-Fi name of the AP after the ESP-AT firmware is downloaded to the ESP32?

• You can use the AT+CWJAP? command to query the default Wi-Fi name. By default, the MAC address is appended to Wi-Fi names.

• AT supports custom Wi-Fi names, which can be set with the following AT commands:

```
AT+CWMODE=1 // set the current device to the softAP mode
AT+CWSAP="SSID","PASSWORD",1,0,4,0 // set softAP parameters
AT+CWSAP? // query the softAP parameters you just set
```

How to use the SPI interface for AT communication?

• The AT firmware provided by Espressif uses UART for communication by default. If you need to use SPI for communication, please configure and compile your own esp-at project. See Compile and develop for more information.

Does the command AT+CWLAP perform an active scan or passive scan in the old version of ESP8266 AT firmware (SDK v1.5.4)?

• The AT firmware that is based on ESP8266_NonOS_SDK v2.2.0 is of version 1.6.2. It supports active scan (default) and passive scan. AT firmware before this version only supports active scan.

How to use AT commands to change the default IP address of the softAP?

• Taking V2.0 and later versions of ESP-AT firmware as an example, the IP address of the softAP can be modified with the following commands:

```
AT+CWMODE=2 // set the current device to softAP mode
AT+CIFSR // query the IP address of the current device's AP
AT+CIPAP="192.168.1.1","192.168.1.1","255.255.255.0" // set the current softAP's IP address
AT+CIFSR // check the newly set softAP's IP address
```
What is the default Bluetooth name for the ESP32 AT firmware?

- The default BLE_NAME for AT firmware is BLE_AT.
- You can use the `AT+BLENAMES` command to query the default Bluetooth name.

How to set the keepalive parameter using the AT+CIPSTART command?

- Example: `AT+CIPSTART="TCP","192.168.1.*",2500,60`

The at_http_webserver example keeps restarting. How to fix it?

This issue has been fixed on the master branch. Please rebase the latest master branch. Or, you can update to `CONFIG_SPI_FLASH_USE_LEGACY_IMPL=y` in the sdkconfig configuration to fix the crash.

Is it possible to set the ESP32-WROOM-32 module to HID keyboard mode with AT commands?

Yes, please refer to Bluetooth LE AT Commands.

How does ESP-AT perform the BQB certification?

- Please refer to Updates to multiple BQB Bluetooth certification options of ESP32 and ESP module certificates.

I am new to ESP-AT firmware. Which AT firmware version shall I choose for ESP8266, NONOS or RTOS

- It is recommended to use the RTOS version, which is being actively maintained now. NONOS is an older AT version.
- The two versions are quite different in terms of logic. Besides, RTOS supports more features and fixes the bugs that exist in NONOS version. RTOS version is now and will be our focus in the long run. We will fix bugs more timely and constantly add new features in this version.
- Please download RTOS AT bin.
Is it possible to set the Bluetooth LE TX for ESP-AT?

- Yes, ESP32 shares an antenna for Wi-Fi and Bluetooth LE. See AT+RFPOWER for more details.

For the AT+CIPTCPOPT command, if multiple clients are connected and disconnected from time to time when ESP32 is the server, should AT+CIPTCPOPT be configured each time?

- You do not need to configure it every time. This setting is for the connection itself, not for the client, that is, you configure link0-link4 and use the socket option of whichever link the client uses.

After I migrate from ESP8266 NONOS AT to RTOS AT (v2.0.0.0 and above), flash the firmware successfully, and start up AT, why no “ready” is returned?

- AT communication pins of the ESP8266 RTOS version have been changed to GPIO13 and GPIO15.
- See Hardware connection for more details.

How to download the AT firmware on Espressif’s official website?

- Download the flash tool: Flash Download Tools.
- See AT Downloading Guide for the download address.

Why is the error “flash read err,1000” printed on the serial port after powering up the newly purchased ESP32-WROVE-B module? How to use AT commands for this module?

- The ESP32 WROVER module is shipped without AT firmware, so the error “flash read err” appears.
- If you want to use the AT command function of ESP32-WROVER-B, please refer to the following links to get the firmware and flash it.
  - Download firmware;
  - Connect hardware;
  - Flash firmware.
How to combine all the bin files compiled by esp-at?

- You can use the combine button of the Flash Download Tools.

After ESP32 enters the passthrough mode using AT commands, can ESP32 give a message if the connected hotspot is disconnected?

- Yes, you can configure it with `AT+SYSMSG`, i.e., set `AT+SYSMSG=4`. In this way, the serial port will report “WIFI DISCONNECT\r\n” when the connected hotspot is disconnected.
- Note that this command is added after AT v2.1.0. It is not available for v2.1.0 and earlier versions.

Do AT commands support IPv6?

- Currently AT does not support IPv6, but only IPv4.

How does ESP8266 get the SNTP time of a half time zone using AT commands?

V2.2.0.0 and later versions of the ESP8266 AT firmware support obtaining SNTP time of a half time zone. Below is an example:

```
AT+GMR
AT+CWMODE=1 // set the device to station mode
AT+CWNJAP="SSID","password" // connect to an AP, enter ssid and password
AT+CIPSNTPCFG=1,530 // configure to obtain the SNTP time of the half time zone 5:30
AT+CIPSNPTIME? // query the SNTP time of the half time zone
```

How to handle special characters in AT commands?

- Please refer to escape character syntax.

How to get the source code of AT firmware?

- ESP-AT firmware is partially open-source. See esp-at for the open-source repository.
Why does the ESP-AT firmware always return the following message after I powered up the device and sent the first command?

ERR CODE:0x010b0000
busy p...

• This message means that the previous command is being executed.
• Normally only “busy p...” is displayed. The ERR CODE is displayed because the error code prompt is enabled.
• If you receive this message after sending the first command on power-up, the possible reasons are: the command is followed by the unnecessary newline/space/other symbols; or two or more AT commands are sent in succession.

Does ESP8266 AT+MQTTPUB support the data in json format?

• Yes, below is an example:

```
AT+CWMODE=1 // set
→ the current device to station mode
AT+CJAP="ssid","passwd" //
→ connect to the specified AP
AT+MQTTUSERCFG=0,1,"ESP32","espressif","1234567890",0,0,"" // set
→ MQTT parameters
AT+MQTTCONN=0,"192.168.10.234",1883,0 //
→ connect to the specified MQTT server
AT+MQTTPUB=0,"topic","{"timestamp":"20201121085253"}"",0,0 //
→ publish a piece of json data to Topic
```

How does the ESP32 AT send data to the debug APP over Bluetooth LE?

• The following commands demonstrate how ESP32 AT sends data to the debug APP over Bluetooth LE:

```
AT+RESTORE // initialize the device
AT+BLEINIT=2 // set ESP32 to SERVER mode
AT+BLEGATTSSRVCRE // GATTS creates services
AT+BLEGATTSSRVSTART // GATTS starts services
AT+BLEADDR? // query the MAC address of the Bluetooth LE device
→
AT+BLEADVSTART // start Bluetooth LE advertising and connect to the device using the APP
→
AT+BLEGATTSCHAR? // query the characteristics that are allowed to notify
→
AT+BLEGATTSNTFY=0,1,6,6 // notify the 6-byte data using the sixth characteristic in the first service
→
then, ESP32 sends data to the APP through the serial port tool, such as "12345"
```
How does the ESP32 module use AT commands to implement encrypted Bluetooth pairing with the static secret key?

- The commands are as follows:

```plaintext
AT+RESTORE // initialize the module
AT+BLEINIT=2 // initialize the module as a server
AT+BLEGATTSRVCRE // GATTs creates services
AT+BLEGATTSRVSTART // GATTs start services
AT+BLEADDR? // query the address of the Bluetooth LE device
AT+BLESECPARAM=1,0,16,3,3 // set Bluetooth LE encryption parameters
AT+BLESETKEY=123456 // set the static secret key for Bluetooth LE pairing
AT+BLEADVSTART // start Bluetooth LE advertising and connect to ESP32 using the APP
AT+BLEENC=0,3 // after the connection is established, use this command to generate an encrypted connection request, and enter the secret key
```

How to use the Ethernet function of the ESP32 AT?

- In terms of hardware, you can use the ESP32-Ethernet-Kit development board to do the test.
- Since GPIO19 and GPIO22 of ESP32-Ethernet-Kit are already occupied, you need to change the default UART pins to other free GPIOs, such as GPIO4 and GPIO2.
- In addition, you need to enable the AT ethernet support function in menuconfig when compiling your esp-at project. By default, this function is not enabled in the AT.bin downloaded from Espressif’s official website.

Why does the ESP32 loaded with the ESP-AT firmware V2.1.0.0 return ERROR after the AT+BLUFI=1 command is sent?

- The ESP-AT firmware V2.1.0.0 does not support BluFi provisioning. To use this feature, you need to compile the latest master (V2.2.0.0) version of esp-at code to generate the firmware.
- When compiling the project, go to menuconfig -> Component config -> AT -> AT blufi command support, enter “Y” to enable this feature.

```plaintext
AT+RESTORE // initialize the device
AT+GMR // check the firmware version
AT+BLUFI=1 // enable the BluFi feature
AT+BLEADDR? // query the device’s address
```
Is it possible to set b/g/n modes for modules using ESP32 AT commands?

- This feature is supported since ESP32 AT v2.1.0.0.
- AT+CWSTAPROTO is used to set and query the 802.11 b/g/n in station mode.
- AT+CWAPPROTO is used to set and query the 802.11 b/g/n in softAP mode.

The ESP32 AT UART1 communication pins do not match the default pins in the datasheet?

- ESP32 supports the GPIO Matrix. When compiling esp-at, you can modify the UART1 pin configuration through the software in menuconfig, so the pins may not match with those in the datasheet.

When the host MCU sends an AT command to the ESP32 device (AT firmware version: V2.1.0.0), there is no response. What is the reason?

- A terminator (“ATrn”) must be added after an AT command when the host MCU sending AT commands to an ESP32 device. Please see Check Whether AT Works.

Using ESP8266 v2.1.0.0 version AT firmware, how to disable the default power save mode?

- The default power save mode can be disable via the AT+SLEEP=0.

Received the following log after sending an AT command. What is the reason?

```
busy p...
OK
```

- Please check whether there are redundant spaces or line breaks in your AT commands, e.g., extra CR and LF. You can also take data from the communication line to see what could be wrong.
- For more information, please refer to AT Command Types.

What is the maximum value of the parameter length of AT+BLEGATTSNTFY and AT+BLEGATTSIND?

**CHIP: ESP32**

- The maximum value of length is MTU - 3, and the MTU can support up to 517 bytes, so the maximum value of length is 514 bytes.
How to enable full calibration mode for ESP8266 NONOS AT firmware?

- The NONOS AT firmware uses partial calibration by default, and full calibration can be enabled in the following way:
  modify file esp_init_data_default_v08.bin, byte[114] = 3 (for partial calibration, byte[114] = 1, while for full calibration, byte[114] = 3).

What is the maximum rate of ESP32 AT BLE UART transparent transmission?

- In an open office environment, when the serial port baud rate is 2000000: the average transmission rate of ESPAT BT is 0.56 Mb, and the average transmission rate of ESPAT BLE is 0.101 Mb.
- In shielding box environment, the data will also be provided after the test finished in the future.

How to get the AT firmware of the ESP32-MINI-1(ESP32-U4WDH inside) module?

- Please refer to How_to_download_the_latest Temporary_version_of_AT_from_github.

How to set ADV broadcast parameters after it exceeds 32 bytes?

:CHIP: ESP32 :


Does AT support Wi-Fi roaming function?

:CHIP: ESP32|ESP32-S2|ESP32-C3

- Not supported.
How to download redirected resources via ESP8266 AT firmware?

- ESP8266 AT HTTP command does not support redirection. After getting the status code 301 (permanent redirection) or 302 (temporary redirection) returned by the server, it will not automatically redirect to the new URL address.
- You can use wireshark or postman to get the actual URL, and then access it through HTTP commands.
  - Please note that ESP8266-IDF-AT_V2.1.0.0 cannot support HTTP command by default, you need to compile AT firmware based on esp-at, please refer to ESP8266 platform. HTTP needs to be enabled in menuconfig: menuconfig -> Component config -> AT -> [*] AT http command support.
  - The maximum length of the parameter URL in AT+HTTPCLIENT is 256. When the length of the actual URL obtained exceeds 256, it will return ERROR. You can use TCP related commands to send the a constructed HTTP request message to obtain the resource.

When using ESP-AT to send TCP data, sometimes the data is messy/partially lost. What should I do?

- It is recommended to add hardware flow control or software flow control:
  - Hardware flow control: CTS and RTS signals
  - Software flow control
- If necessary, you can add some logic to handle errors in the code. For example, when your device accidentally entered the transparent transmission mode, or there is error transmission in the transparent transmission mode, send +++ in time to exit the transparent transmission, and resend the AT+CIPSEND command.

When ESP32 performs BLE OTA, it connects to phone via BLE and connects to MCU via UART, then performs OTA to MCU. But the data transmission between ESP32 and MCU is low even after increasing MCU via phone. Where should I check for such issue?

- The reason may be that ESP32 and the mobile phone did not set the MTU successfully, or the UART limitation of ESP32 and MCU communication. Therefore, it is recommended to check/improve from the following points:
  - Connection between ESP32 and mobile phone
    1. Only the BLE client supports setting the GATT MTU length, and the BLE connection needs to be established before the MTU length is set. The final actual MTU length needs to be negotiated. If it returns OK, it only means the negotiation process is triggered. Therefore, the length you set before may not be valid. It is recommended to use the query command AT+BLECFGMTU? to check the actual MTU length after setting.
    2. Use BLE SPP, the BLE transparent transmission mode, to increase the transmission rate.
  - Connection between ESP32 and MCU: increase the baud rate of UART appropriately to increase the transmission rate.
When using ESP32-C3 as a Server with AT firmware version v2.2.0.0, what is the maximum number of connections allowed by the AT+CIPSERVERMAXCONN command?

- The maximum number of connections allowed to be established by the AT+CIPSERVERMAXCONN command is 5 by default.
- You can configure the “menuconfig -> Component config -> AT -> Socket Maximum Connection” parameter to allow more connections.
- If you need to support more than 10 connections, you also need to increase the “menuconfig -> Component config -> LWIP -> Max number of open sockets” (default is 10) configuration.
- However, the maximum number of connections allowed during actual operation depends on the remaining available memory of the chip. When no more connections can be established, it is recommended to use the AT+SYSRAM command to query the current remaining available memory.

When using the release/v2.1.0.0 version of the AT firmware, what is the maximum number of BLE devices that ESP32 supports to save binding and paring information for?

- Up to 15 BLE devices.

The maximum length of AT+BLEADVDATA broadcast data is 31. How to realize a bigger data length support?

**CHIP: ESP32**

What authentication methods does WPA2 Enterprise support?

**CHIP: ESP8266 | ESP32 | ESP32-C3**
- Only EAP-TLS/EAP-PEAP/EAP-TTLS are supported. For details, please refer to the AT+CWJEAP <https://docs.espressif.com/projects/esp-at/en/latest/AT_Command_Set/Wi-Fi_AT_Commands.html#esp32-only-at-cwjeap-connect-to-a-wpa2-enterprise-ap>_ command introduction.
Are there any AT+HTTPCPOST usage examples?

**CHIP: ESP8266**

- While compiling firmware manually under the master version of ESP-AT, it is required to change the “at process task stack size” to over 4096 in menuconfig, the specific operation steps are as follows:
  - `./build.py menuconfig—–>AT—–>(5120)`. The stack size of the AT process task in AT library, which will be used to process AT command.

  - [*] AT http command support:

    ```
    AT+CWMODE=1 //set as station mode OK
    AT+CWJAP="iot","123456789" WIFI CONNECTED WIFI GOT IP
    AT+HTTPCPOST="http://61.172.47.198:8082/hello/test",172 OK
    >AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    SEND OK
    ```

Are there any examples of using AT+CIPRECVDATA to receive cache data from the server?

**CHIP: ESP8266**

```
AT+CWMODE=1 //set as station mode OK
AT+CWJAP="iot","123456789" WIFI CONNECTED WIFI GOT IP
AT+CIPSTART="TCP","192.168.3.129",8080 CONNECT OK
AT+CIPRECVMODE=1 OK //The server send 16bytes data to client
AT+CIPRECVLEN? //Inquire server cached data
+CIPRECVLEN:16 //Ensure the quire length isn't zero, otherwise, the data received will return ERROR.
AT+CIPRECVDATA=1080
```

I use ESP32 AT firmware to send BLE scan command, but the scan response packet is not received. What is the reason?

- The AT commands used are as follows:

  ```
  AT+BLEINIT=1
  AT+BLESCANPARAM=0,0,0,100,50
  AT+BLESCAN=1
  ```

  There is a broadcast packet, but there is no scan response packet; the log of the command reply is shown in the figure:
Is there a limit to the maximum length of the broadcast packet sent by the “AT+BLEADVDATA” command?

- The maximum length of broadcast packet by “AT+BLEADVDATA” is 31 bytes.

What is the maximum value of the length parameter in the AT+BLEGATTCWR command?

- The maximum value of length in the AT+BLEGATTCWR command is corresponds to the val_max_len parameter setting under the example.csv file, and it is recommended not to exceed 512. Please refer to the description in README.

When using the v2.2.0.0 version of AT firmware to connect ESP32 to AP, it will automatically connect to the AP again after being reset and powered on. How to cancel this setting?

- This can be realized by using the AT+SYSSTORE=0 command. After calling this command, the related configuration information of affected commands will not be saved to flash. That is to say, before connecting AP using the AT+CWJAP command, you can call AT+SYSSTORE=0 first to make old AP information not be stored to flash.
Does ESP32-AT support PPP?

- Not supported, please refer to pppos_client demos for your own implementation.

How to enable Wi-Fi Debug for AT?

CHIP: ESP8266 | ESP32 | ESP32-C3 | ESP32-S2

- Enable log level: . /build.py menuconfig -> Component Config -> Log output -> Default log verbosity set to Verbose.
- Enable Wi-Fi debug: . /build.py menuconfig -> Component config -> Wi-Fi -> Enable WiFi debug log -> Wi-Fi debug log level (Wi-Fi Debug log Verbose).

Where should I pay attention to when updating certificates using the AT+SYSFLASH command?

- The certificate length must be 4-byte aligned.
- The certificate bin needs to be generated via tools/AtPKI.py, please refer to the example in esp-at/tools/README.md. For instance:
  ```
  python AtPKI.py generate_bin -b mqtt_cert_v2.bin cert mqtt_client.crt
  ```

What's the default type of content-type in AT+HTTPCPOST command?

CHIP: ESP8266 | ESP32 | ESP32-S2 | ESP32-C3

- The default type is application/x-www-form-urlencoded.

Is there a length limit on data sent with the AT+HTTPCLIENT command?

- Yes, the total length of data set with this command cannot exceed 256 bytes. This is due to the data is stored in command parameters. If you are going to send data with bigger length, it is recommended to configure via the AT+HTTPCPOST command, or use the TCP command to emulate Http to send data.
What TLS versions are supported by AT?

- TLS 1.0, TLS 1.1, TLS 1.2 are supported, the exact version numbers can be found in menuconfig-->Component config-->mbedTLS.

ESP-WIFI-MESH development framework

What is the maximum data transmission load for Wi-Fi mesh?

- Up to 1456 bytes.

Does ESP32's Wi-Fi Mesh supports No Router self-networking?

- Yes, please refer to example esp-mdf/examples/function_demo/mwifi/no_router.

What is the maximum number of node layers allowed when ESP32 uses Wi-Fi Mesh?

- In the Wi-Fi Mesh network, you can set the maximum number of layers via esp_mesh_set_max_layer().
- For tree topology structure, the maximum number is 25; while for chain topology structure, the maximum number is 1000.

When using an ESP32 development board to test the esp-mdf/examples/function_demo/mwifi/router example, After ESP32 is connected to the router, the device name in connection is “espressif”. How to modify this name?

- Please modify the “menuconfig → Component config → LWIP → (espressif) Local netif hostname” setting.

Can Wi-Fi Mesh send messages to specific nodes via TCP Server?

- Wi-Fi Mesh network can send data to the specified node or group in the TCP server, please refer to the demo.
During the operation of the ESP32 Wi-Fi Mesh network, if the Root node is lost, what events will the system report back?

- If the Root node is lost, all nodes will trigger ‘MDF_EVENT_MWIFI_PARENT_DISCONNECTED (MESH_EVENT_PARENT_DISCONNECTED)’, and then start rescanning and re-election until a new Root node is elected.

I’m using ESP32 for Wi-Fi Mesh application with the `esp_mesh_send()` function, but the server did not receive any data. How to transfer data from leaf nodes to external servers?

- `esp_mesh_send()` can only be used for data communication within the Wi-Fi Mesh network.
- If leaf nodes want to send data to an external server, the data needs to be forwarded through the root node.
- The correct approach is: the leaf node first sends the data to the root node, and the root node then sends the data to the external server.

How do I upgrade my ESP-MESH device via OTA after networking?

- The ROOT node can connect to the server to get the upgrade bin file and then send the firmware to the corresponding module via MAC address for OTA upgrade.
- For more information, please refer to `mupgrade demo`.

Can you provide ESP-MESH light reference design?

- The overall design of the lamp is done by a third-party factory and we do not have a schematic or PCB layout. But from the module level, we only need to supply power to the chip and the chip outputs PWM to control the color or color temperature change of the lamp, which does not involve complicated design.
- Please refer to `ESP-MDF` for more information on MESH.

What is the default mode for ESP-MESH nodes without any configuration?

- The default is IDLE mode.
ESP-MESH starts with AP+STA mode enabled, can the phone search for APs?

- No, ESP-MESH is a private protocol of Espressif, please refer to WIFI-MESH Introduction.

Do I need to rescan for all the newly added devices when the original device has already been networked?

- No, just scan through the current child nodes and find the one with the strongest signal as its parent node.

BLE Mesh development framework

<table>
<thead>
<tr>
<th>[]</th>
</tr>
</thead>
</table>

What is the maximum data transmission load for Bluetooth® LE (BLE) mesh?

- Up to 384 bytes for the single packet in application layer, up to 11 bytes in the bottom layer with no sub-packages.

Could you provide an example of networking through ESP32 BLE-Mesh? What APP can be used for BLE-Mesh networking?

- Please use example onoff_server and use nRF Mesh APP for mobile phones.
- For the network configuration process, please refer to Getting Started with ESP-BLE-MESH.

For unprovisioned device in BLE-MESH, the default name is ESP-BLE-MESH, how to modify this name?

- You can use API esp_ble_mesh_set_unprovisioned_device_name(), it is suggested to call it after esp_ble_mesh_init(), otherwise the device name is still ESP-BLE-MESH.

Audio development framework

| [] |
What is the maximum power of supported speakers for ESP32 series audio development board?

- ESP32 development board uses NS4150 PA by default, and its maximum power is 3 W according to its datasheet.

Does Alexa solution have certain requirements for environmental noise?

- The current Espressif voice solution can meet the environmental requirements of a signal-to-noise ratio of less than 5dB, and for some fixed noise scenarios, it can also be less than 0dB (need to be optimized for the actual product).

There is an AUX input on the ESP32 AI development board, can MIC be used to pick up the sound?

- The ESP-ADF development framework can choose a variety of ways to pick up sound, including MIC input and Line-in.
- The pick-up method is as follows:

```c
typedef enum {
    AUDIO_HAL_CODEC_MODE_ENCODE = 1, /*! <select adc */ // MIC pickup
    AUDIO_HAL_CODEC_MODE_DECODE, /*! <select dac*/
    AUDIO_HAL_CODEC_MODE_BOTH, /*! <select both adc and dac */ // MIC +
    AUDIO_HAL_CODEC_MODE_LINE_IN, /*! <set adc channel */, //
    AUDIO_HAL_CODEC_MODE_MICROPHONE //microphone pickup
} Audio_hal_codec_mode_t;
```

- The configuration of the pickup method is as follows:

```c
audio_board_handle_t board_handle = audio_board_init();
audio_hal_ctrl_codec(board_handle->audio_hal, AUDIO_HAL_CODEC_MODE_DECODE, AUDIO_HAL_CODEC_MODE_LINE_IN); //To MIC pickup, please modify this configuration.
```

When using ESP32-WROVER-B module + ES8311 to design audio development board, which pins can be selected for MCLK clock?

- On the hardware side, MCLK can only use GPIO0, GPIO1, and GPIO3 pins. Other pins cannot be used. You can read ESP32 Datasheet about CLK_OUT* pins in IO_MUX table. GPIO0 is used by default.
- Please refer to the schematic of ESP32-LyraT-Mini.
- For allocation of pins, please refer to ESP32-LyraT-Mini V1.2 Hardware Reference.
Can ESP32-WROVER-E module use one I2S line to realize simultaneous broadcasting and recording?

- Yes, you can refer to ESP32-LyraT Development Board.

Do Espressif modules support Spotify Connect?

:CHIP: ESP32 | ESP32-S2 | ESP32-S3 :

- Not supported yet. It is suggested to try dlna, which has similar functions.

When running the korvo_du1906 example on an ESP32-Korvo-DU1906 board, a reboot caused the following error message: *Guru Meditation Error: Core 0 panic’d (IllegalInstruction). Exception was unhandled.* How to resolve such issue?

- Please check the power supply.
- It is recommended that the system be connected to an at least 5 V/2 A power adapter for sufficient current supply.

Can ESP-DSP fft run 4096, 8192 and more samples?

- Yes, up to 32 K samples are supported. The maximum number can be configured in menuconfig, e.g., for fft demo, go to idf.py menuconfig-->Component config-->DSP Library-->Maximum FFT length-->(*32768).

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.
Does ESP Azure library support Azure IoT Central? Is there a demo?

- ESP Azure already supports Azure IoT Central. But there is no relevant example on the master.
- The PnP example on the ESP Azure’s preview/pnp_example branch will report some actual data from sensors, you can refer to the operation of Azure IoT Central for the data management.

What should I do to connect ESP32 to Alibaba Cloud via ESP32 + Ethernet + MQTT?

- Use esp-aliyun but replace the Wi-Fi initialization code with Ethernet initialization. You can refer to the Ethernet example under ESP_IDF.

what do Alexa LED states indicate?

- You can refer to Alexa LEDs.

Community sw and platforms

- ios application
- Android application
- Camera application

What type of camera does the ESP32 series chip support?

- Please refer to Camera models supported by ESP32 series.
Where is the factory firmware of ESP-EYE?

- Please refer to ESP-EYE’s factory firmware.

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 8 transmit power levels, corresponding to -12 ~ 9 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?

<table>
<thead>
<tr>
<th>Current</th>
<th>MAX (mA)</th>
<th>Min (mA)</th>
<th>Average (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advertising: Adv Interval = 40 ms</td>
<td>142.1</td>
<td>32</td>
<td>42.67</td>
</tr>
<tr>
<td>Scanning: Scan Interval = 160 ms, Window = 20 ms</td>
<td>142.1</td>
<td>32</td>
<td>44.4</td>
</tr>
<tr>
<td>Connection(Slave): Connection Interval = 20 ms, latency = 0</td>
<td>142.1</td>
<td>32</td>
<td>42.75</td>
</tr>
<tr>
<td>Connection(Slave): Connection Interval = 80 ms, latency = 0</td>
<td>142.1</td>
<td>32</td>
<td>35.33</td>
</tr>
</tbody>
</table>

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 menu-config:

- 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 -x(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_t type);.

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 Blu-Fi. So it is recommended to use Blu-Fi for networking.

What is the operating current for ESP32 Classic Bluetooth®

A2DP (Single core CPU 160 MhzDFS = falsecommit a7a90f)

<table>
<thead>
<tr>
<th>Current</th>
<th>Maximum (mA)</th>
<th>Minimum (mA)</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scanning</td>
<td>106.4</td>
<td>30.8</td>
<td>37.8</td>
</tr>
<tr>
<td>Sniff</td>
<td>107.6</td>
<td>31.1</td>
<td>32.2</td>
</tr>
<tr>
<td>Play Music</td>
<td>123</td>
<td>90.1</td>
<td>100.4</td>
</tr>
</tbody>
</table>

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_BT M: 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 slot 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:

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

When using the “Blufi” example to configure network for ESP32, the Wi-Fi cannot be connected during the distribution process via the EspBluFi application since a wrong Wi-Fi has been configured. Then the device is restarted after sending a SCAN command from the application. What is the reason?

• The “Blufi” example stipulates that Wi-Fi “SCAN” commands cannot be sent when Wi-Fi is connected.
• To solve this issue, you can add ESP_ERROR_CHECK(esp_wifi_disconnect()); to the first line of the ESP_BLUFI_EVENT_GET_WIFI_LIST(); function under the blufi_example_main.c file.
Using ESP32, how to specify a BLE connection/transmit operation to run on core 0?

- Currently, ESP32’s BLE connection/transmit operation only can be run on core 1. You can enable this via “menuconfig -> Component config -> FreeRTOS -> Run FreeRTOS only on first core (enable this option)”.
- According to this application requirement, you can distribute tasks to a certain core using the “xTaskCreatePinnedToCore()” or “xTaskCreateStaticPinnedToCore()” API. For specific instructions, please see core assignment.

When setting name for the bluetooth of an ESP32 device using Chinese characters, messy code shows instead. What is the reason

- This is because the Chinese encoding format of the editor is not UTF-8 at this time, and the encoding format of the editor needs to be changed to UTF-8.

Using ESP32, when uploading sub-packages on the Bluetooth channel, the maximum transmission data length of a packet is 253 (MTU is set to 263), which results in slower transmission when a large number of data packets are transmitted for multi-packet reading. Is there a Blufi extension protocol that can support the transmission of a larger length of data in one packet, or are there other solutions to increase the transmission rate?

- When transmitting a large number of data packets on the Bluetooth channel for multi-packet reading, the transmission is slow, and the transmission speed can be improved by adjusting the Bluetooth connection parameters.
- The BLE packet length setting depends on the ESP_GATT_MAX_MTU_SIZE setting, please refer to the Description.
- The configured MTU size will affect the data transmission rate. The effective MTU length needs to be changed by MTU exchange to change the default MTU size. The MTU size used in the final MTU exchange is used as the MTU size for the communication between the two devices. You can check the value of the MTU after exchange, such as the follows:

```c
case ESP_GATTS_MTU_EVT:
    ESP_LOGI(GATT_TAG, "ESP_GATTS_MTU_EVT, MTU%d", param->mtu.mtu);
```

What profile does ESP32’s classic Bluetooth® support?

- It supports A2DP, AVRCP, SPP, HFP currently, while HID is under development.
How many stable connections can be reached for ESP32-C3’s Bluetooth® LE (BLE)?

- Eight.

How to adjust the BLE advertising interval?

- The advertising interval is decided by `adv_int_min` and `adv_int_max` parameters in BLE advertising struct, which configures the minimum and maximum advertising interval respectively.
- The range of advertising interval value is 0x0020 to 0x4000 and the default value is 0x0800. The interval time is the value * 0.625 ms, i.e., 20 ms to 10.24 sec.
- If the values of `adv_int_min` and `adv_int_max` are different, the advertising interval is within the range of the two values. If the values are the same, the interval will be this fixed value.

How to input the PIN code via mobile phone during ESP32’s Classic Bluetooth Pairing mode?

You can disable Secure Simple Pairing to support only Legacy Pairing.

- From `esp-idf` v3.3 to v4.0 (not include v4.0): Component config > Bluetooth > Bluedroid Enable > [ ] Classic Bluetooth > [ ] Secure Simple Pairing
- esp-idf v4.0 and above: Component config → Bluetooth → Bluedroid Options → [ ] Secure Simple Pairing

How much memory does ESP32 Bluetooth occupy?

- Controller:
  - BLE single mode: 40 KB
  - BR/EDR single mode: 65 KB
  - Dual mode: 120 KB
- Main equipment:
  - BLE GATT Client (Gatt Client demo): 24 KB (.bss+.data) + 23 KB (heap) = 47 KB
  - BLE GATT Server (GATT Server demo): 23 KB (.bss+.data) + 23 KB (heap) = 46 KB
  - BLE GATT Client & GATT Server: 24 KB (.bss+.data) + 24 KB (heap) = 48 KB
  - SMP: 5 KB
  - Classic Bluetooth (Classic Bluetooth A2DP_SINK demo, including SMP/SDP/A2DP/AVRCP): 48 KB (.bss+.data) + 24 KB (heap) = 72 KB (an additional 13 KB is added when the example is running)
Note: The above heap (Heap) all include the task stack (Task Stack), because the task stack is allocated from the heap and counted as a heap.

- **Optimize PSRAM version:**
  
  In ESP-IDF v3.0 and later versions, if you open the PSRAM related options of the Bluetooth menu in menuconfig, and put part of the .bss/.data section and heap of Bluedroid (Host) into PSRAM, this can save additional 50 KB of memory spaces.

---

When using the “gattc_gatts_coex.c” example on ESP32 to test BLE multi-connection, it can only connect to 4 devices even after I set the “BLE Max connections” in “menuconfig” to 5. What is the reason?

- Please set the “BT/BLE MAX ACL CONNECTIONS” in “menuconfig” to 5.

---

Does ESP32-C3 BLE support master and slave mode at the same time? What is the number of connections in master mode and slave mode?

**IDF: release/v4.3, master**

- ESP32-C3 supports master and slave mode at the same time, sharing 8 connections. For example, if ESP32-C3 connects to 4 slave devices, it can be connected by 8 - 4 = 4 master devices.
- In addition, when ESP32-C3 is used as a slave, it can be connected by 8 master devices; when used as a master, it can connect to 8 slave devices.

---

What is the maximum MTU Size of ESP32 Classic Bluetooth?

- ESP32 Classic Bluetooth has two protocols, namely A2DP and SPP. The maximum MTU Size setting of BT A2DP (default) is 1008 bytes, of which the header occupies 12 bytes and the actual amount of data transmitted by the application layer is 1008 - 12 = 996 (bytes); the maximum MTU Size of BT SPP (default) Set to 990 bytes.

---

How to resolve the frequently occurred ELxXX error (such as ELx200) when Wi-Fi and Ble co-exit

**CHIP: ESP32**

- It has been fixed in commit 386a8e37f19f8ec9ef62ef72441e6e1272fa985b9. Please switch to the corresponding commit to test.
How does BLE capture packets?

- There are many tools available in the market, such as:
  - TI Packet sniffer
  - NRF Packet sniffer

When using an ESP32 development board to test several versions of bluefi example under ESP-IDF for networking, the following error kept printing. What is the reason?

```
E (117198) BT_L2CAP: 12ble_update_att_acl_pkt_num not found p_tcb
W (117198) BT_BTC: btc_blufi_send_encap wait to send blufi custom data
```

- When this error occurs, please modify the `esp_ble_get_cur_sendable_packets_num(blufi_env.conn_id)` to `esp_ble_get_sendable_packets_num()` in the `components/bt/host/bluedroid/btc/profile/esp/blufi/blufi_prf.c` file.
- This bug has been fixed in all branches, you can update ESP-IDF to the latest release version.

When using ESP32, can light-sleep mode be enabled for Bluetooth and can Bluetooth be kept connected in light-sleep mode?

- To use light-sleep mode for ESP32, release/4.0 or above versions of ESP-IDF and a 32.768 kHz crystal oscillator are needed.
- Bluetooth can be kept connected in light-sleep mode. Please refer to Bluetooth modem sleep with external 32.768 kHz xtal under light sleep.

How to modify the Bluetooth broadcast name of ESP32?

- The structure to be modified is as follows:

```c
static uint8_t raw_adv_data[] = {
  /* flags*/
  0x02, 0x01, 0x06,
  Tx power*/
  0x02, 0x0a, 0xeb,
  /* service uuid*/
  0x03, 0x03, 0xFF, 0x00,
  /* device name*/
};
```

(continues on next page)
• The above /* device name */) is the modified item. Among them, 0x0f is the total length of the field type plus specific content, and 0x09 indicates that this type refers to the device name. Subsequent 'E', 'S', 'P', ' ', 'G', 'A', 'T', 'S', ' ', 'D', 'E', 'M', 'O' are the ASCII code expressions of the broadcast device name.

**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 0x0007(mask 0xffff). Current value 0xffff
E (2329) emac: Initialise PHY device Timeout
ESP_ERROR_CHECK failed: esp_err_t 0xffffffff (ESP_FAIL) at 0x40084140
```

(continues on next page)
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:

```c
{ ... 
    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";
    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.

1.1. Question search
When using ESP32 to design a self-developed Ethernet board, after downloaded the official esp- idf/examples/ethernet example, errors are reported as follows, what is the reason?

- This error indicates something is wrong with your hardware circuit. The RMII clock is not working normally with the PHY, causing the PHY failed to read registers. For the more information about RMII clock, please refer to Instructions.

**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 ESP-WIFI-MESH and Bluetooth® LE Mesh?

No.

However, the ESP32 supports coexistence between ESP-WIFI-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 Driver esp-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 peripherals SARADC1 SARADC2 AMPHALL 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 \( \text{LEDC}_{\text{HSTIMERx\_DUTY\_RES}} \) or \( \text{LEDC}_{\text{LSTIMERx\_DUTY\_RES}} \).
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 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.

---

**Does the ESP-IDF SDK USB interface support HID and MSC modes?**

- Our SDK will provide examples of HID and MSC classes in the future. And specific device classes need to be implemented by themselves.

---

**When using DAC output for ESP32-S2-Saola-1, the power supply is 3.3 V. But the actual tested voltage is only 3.1 V. Why?**

- Due to the internal voltage drop, even when using 3.3 V power supply, the actual maximum output is only about 3.2 V.
If I float the ADC pin and print out VDD3P3 value (65535), then the voltage of VDD3P3 should be 65535/1024 = 63 V. Why this is not the correct voltage value?

- ADC pins cannot be left floating, and the value measured by floating ADC pins is not the correct value.

When using ESP8266 to generate PWM by directly writing to the register of the hardware timer FRC1, I found there are error PWM outputs after Wi-Fi is initialized since it may disturb the interrupt of FRC1. Is it possible to use FRC2 instead to generate PWM? Or is it possible to set FRC1 higher priority than Wi-Fi?

- FRC2 cannot be used as it is occupied by the system. Wi-Fi uses NMI interrupt, which have a higher priority than other ordinary interrupts. It is recommended to use the PWM library of ESP8266_RTOS_SDK. Please refer to ESP8266_RTOS_SDK/examples/peripherals/pwm example.

I’m using v3.3.3 version of ESP-IDF to test the ledc example on ESP32. The LED PWM outputs when Auto Light Sleep mode is disabled, but does not output when this mode is enabled. According the description of LED PWM in ESP-IDF programming guide, LED PWM should work in sleep modes. What is the reason?

- v3.3.3 does not support LED PWM working in sleep modes. Please use the ledc example under the new versions of ESP-IDF (v4.0 and later versions) to test, e.g., ESP-IDF release/v4.2 version of the SDK. Plus, it is also necessary to change the LED PWM clock source to the internal RTC_8M clock source. Please see below:

```c
ledc_timer_config_t ledc_timer = {
    .duty_resolution = LEDC_TIMER_13_BIT,
    .freq_hz = 5000,
    .speed_mode = LEDC_LOW_SPEED_MODE,
    .timer_num = LEDC_TIMER_0,
    .clk_cfg = LEDC_USE_RTC8M_CLK,
};
```

What is the input resistance of ESP32 ADC?

- ADC is capacitive and can be considered as a large resistance.
When using ESP32’s ADC to detect the power supply voltage, is it necessary to divide the voltage?

- Yes, it is necessary if you are using ADC for ESP32 devices to detect voltage. The ADC default reference voltage of ESP32 is 1100 mV. However, the ADC reading width can be expanded by calling the ADC attenuation function `adc_atten_t()`. Please refer to the ADC attenuation configuration instructions as follows:

```c
typedef enum {
    ADC_ATTEN_DB_0 = 0, /*!< No input attenuation, ADC can measure up to approx. 800 mV. */
    ADC_ATTEN_DB_2_5 = 1, /*!< The input voltage of ADC will be attenuated, extending the range of measurement to up to approx. 1100 mV. */
    ADC_ATTEN_DB_6 = 2, /*!< The input voltage of ADC will be attenuated, extending the range of measurement to up to approx. 1350 mV. */
    ADC_ATTEN_DB_11 = 3, /*!< The input voltage of ADC will be attenuated, extending the range of measurement to up to approx. 2600 mV. */
    ADC_ATTEN_MAX,
} adc_atten_t;
```

The maximum data transmission of ESP32 SPI DMA is 4092 bytes. Is it because of hardware limitation?

- Yes. A single node can only store 4092 bytes of data, but the DMA can send more data through link lists.

What is the stable current output for ESP32-S2’s USB interface?

- ESP32-S2 supports USB 1.1 Full-Speed mode, under which the output of data line D+ and D- is voltage signal. Thus, there is no need to consider current driving capability here. As for the driving capability for VBUS line, it has nothing to do with ESP32-S2 as it is decided by the power-supply chip.

Does ESP32-S3’s USB peripheral supports USB Host?

- Yes, regarding this function, ESP32-S3 is the same as ESP32-S2.

Does ESP32-C3 USB support USB serial port function and USB JTAG function?

- Yes.
What reference drivers does ESP32 touch screen have?

- Code: please refer to touch_panel_code.
- Documentation: please refer to touch_panel_doc.

The SPI of ESP32-S2 accesses three SPI Slave devices at the same time, do I need to synchronize the semaphore to access it?

- The same SPI peripheral, as the master, can only communicate with one slave at a time, and CS decides which slave to communicate with. If you connect 3 slave devices to the SPI driver and communicate with them separately, it is okay and recommended.
- You can use the spi_device_transmit() API, which is a blocking interface and returns after a transmission is completed. If there are multiple tasks, you can call this function one by one and use different handles to communicate.

When using an ESP32 board for development and testing based on ESP-IDF release/v4.3, I received the following error log during compilation. What is the reason?

```
spi_flash:Detected size(8192K) smaller than the size in the binary image.
→header(16384K).Probe failed.
```

- The reason is that the configured “Flash Size” is larger than the actual “Flash Size”. In order to avoid misuse of a larger address space, the actual “Flash Size” is checked.

What is ESP32’s highest sampling rate in ADC DMA mode?

- ESP32 supports up to 2 MHz of sampling rate theoretically.

When an ESP32 calling “adc2_get_raw()” between “esp_wifi_start()” and “esp_wifi_stop()”, the read operation fails. What is the reason?

- Because ADC2 is shared with the Wi-Fi module. The Wi-Fi driver uses ADC2 and has higher priority. Therefore, the application can only use ADC2 when the Wi-Fi driver is not activated.
What is the maximum resolution supported by ESP32 LCD? What is the corresponding frame rate?

- ESP32’s LCD can support up to 800 × 480 of resolution, and the corresponding frame rate is about 30 frames. Please see Screen.

Using ESP-WROOM-02D module, can GPIO0, GPIO15, GPIO1 and GPIO3 be used as normal GPIOs?

- Strapping pins (GPIO0 and GPIO15) and download pins (GPIO1 and GPIO3) can be used as normal GPIOs.
- When using the strapping pin as a normal GPIO, you need to pay attention to the level of the strapping pin in the Flash download mode.

What are the USB features of ESP32-S2 and ESP32-S3?

- ESP32-S3 and ESP32-S2 support USB 1.1 OTG, and both support Host and Device functions. On top of that, ESP32-S3 also supports USB-Serial-JTAG peripheral, which can be used to download and debug firmware.

Are there any references to the library and demo of ESP32-S2 USB Host?

- This part is already under internal development and is expected to be released with SDK release/v4.4. If you want to do some functional verification first, please refer to the USB example in esp-iot-solution.

The USB protocol supported by ESP32-S2 is OTG 1.1, with the maximum speed of 12 Mbps. Can it communicate with USB 2.0 devices?

- Most USB 2.0 devices can backward compatible with USB 1.1, so they can communicate with USB 1.1 (in full speed mode).

Does ESP32-S2 support USB camera?

- Yes, but currently ESP32S2 only supports USB 1.1. So please choose the camera which is compatible with USB 1.1. For demo code, please refer to example uvc_stream.
Is there any reference for the example of using ESP32S2 as a USB flash drive (MSC DEVICE)?

• Please refer to `usb_msc_wireless_disk` demo. The average read and write speed currently tested is: read 540 KB/s, write 350 KB/s.

As ESP32-C3 already has USB function, can I download firmware directly via USB without using the cp2102 chip?

• Yes, ESP32-C3 can download firmware via USB. The USB serial port number should be displayed as COMx on Windows devices and ttyACMx on Linux devices.

Does ESP32-C3 support USB Host?

• No, it only supports USB-Serial-JTAG function.

Does ESP32-S2 have USB UVC demo?

• Please refer to `uvc_stream` demo.

Does ESP32 support using ADC2 and Bluetooth simultaneously?

• Yes.

Can ESP32 read SD card whose file format is exFAT?

• Is not supported by default. exFAT is not free and needs license payment to Microsoft (or some other IP provider).
• There are 2 options if the user wants to use exFAT:
  – keep using Fatfs, which is already included in IDF, and pay royalties to Microsoft for the license. The easiest way to do this is through a reseller, for example, exfat-royalties. In IDF, the user needs to modify `ffconf.h` to enable `FF_FS_EXFAT` option.
  – use a commercial third-party FAT implementation, which will include exFAT royalties into its price. Options include hcc-embedded and embedded-access. Some porting to ESP-IDF may be required, but most likely it won’t be very difficult.
What is the maximum transmission speed supported by SPI slave?

:CHIP: ESP32 :
  • ESP32 can support up to 10 M of transmission speed when serves as an SPI slave.

When using ESP32 as an SPI Master device, how many bytes of data can be transferred at one time in non-DMA mode?

  • Up to 64 Bytes of data can be transferred at one time in such condition.
  • But when the transmitted data exceeds 32 bits, you need to set the buffer for SPI data transmission, please refer to the description in SPI Master Driver.
  • When using ESP32 as an SPI Master device to transmit more than 32 bits of SPI data in non-DMA mode, please refer to the example lcd.

When using the ESP32-S3-WROOM-1 (ESP32-S3R2) module to enable its PSRAM configuration based on the “hello-world” example in ESP-IDF v4.4, the following error is printed. What is the reason?

```
E (232) spiram: Virtual address not enough for PSRAM!
```

  • ESP32-S3R2 chip integrates a 4-wire 2 MB PSRAM, please set PSRAM Mode to Quad mode in menuconfig before your action as follows:
    menuconfig → Component config → ESP32S3 Specific → Support for external, SPI connected RAM → SPI RAM config → Mode (QUAD/OCT) of SPI RAM chip in use (Quad Mode PSRAM)

When using the ESP32-S3-WROOM-2 (ESP32-S3R8V) module to enable the PSRAM configuration based on the “hello-world” example in ESP-IDF v4.4, the following error is printed. What is the reason?

```
E (453) psrm: psrm ID read error: 0x00ffff
E (454) cpu start: Failed to init external RAM!
```

  • ESP32-S3R8V chip integrates a 8-wire 8 MB PSRAM, please set PSRAM mode to Octal mode in menuconfig before your action as follows:
    menuconfig → Component config → ESP32S3 Specific → Support for external, SPI connected RAM → SPI RAM config → Mode (QUAD/OCT) of SPI RAM chip in use (Octal Mode PSRAM)
When using ESP32-C3 to drive the LCD display through the SPI interface, can I use RTC_CLK as the SPI clock to make the LCD screen display static pictures in Deep-sleep mode normally?

- Deep-sleep mode: The CPU and most peripherals will be powered down, and only the RTC memory is working. For more information, please refer to the Low Power Management section in ESP32-C3 Datasheet.
- The SPI of ESP32-C3 only supports two clock sources: APB_CLK and XTAL_CLK. RTC_CLK is not supported. Therefore, in the Deep-sleep mode, the LCD screen cannot display static pictures. For more information, please refer to the Peripheral Clock section in ESP32-C3 Technical Reference Manual.

What is the frequency range supported by the ADC DMA mode of the ESP32-S2 chip?

- Frequency limit: 611 Hz ~ 83333 Hz

The ESP32-C3 chip can use USB to download firmware, but it is not supported under ESP-IDF v4.3. How to use USB to download firmware?

- You need to compile under ESP-IDF v4.4 or later versions. After pulling the latest branch and updating the IDF tool, you can compile normally and download it using USB. Please refer to usb-serial-jtag-console for the usage.

Does the ADC of ESP32 support simultaneous sampling of multiple channels?

- No. If you are using ADC to do multi-channel sampling, please implement it via ADC polling scheme.

Does the ESP32 chip support USRAT (Universal Synchronous Asynchronous Receiver Transmitter)?

- It’s not support. ESP32 only supports UART and cannot provide the synchronous clock.

When using the ESP32-WROVER-B module with release/v4.2 version of ESP-IDF, I set the GPIO as an ADC interface, and then set GPIO to other IO mode while with IO mode not effective without any hardware reset, this GPIO does not respond. How do I release the corresponding GPIO mode?

- Please do not set the ADC interface as input-only GPIO.
- When disabling the ADC interface mode, please use adc_digi_stop() to disable the ADC.
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:
  - iOS APP https://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.

After calling esp_netif_t* wifiAP = esp_netif_create_default_wifi_ap() for ESP32-S2 chips, a following call of esp_netif_destroy(wifiAP) to deinit caused a 12-byte of memory leakage. What is the reason?

- It is necessary to call esp_wifi_clear_default_wifi_driver_and_handlers(wifiAP) before esp_netif_destroy(wifiAP). This is the correct deinit process. Following this process, there will be no memory leakage.

When ESP32 & ESP8266 are used as TCP Servers, how can the ports be used again immediately after they are released?

- After closing the TCP socket, it often enters the TIME-WAIT state. At this time, the socket with the same source address of the same port as before will fail. The socket option SO_REUSEADDR is needed. Its function is to allow the device binding to be in TIME-WAIT state, the port and source address are the same as the previous TCP socket.
- So the TCP server program can set the SO_REUSEADDR socket option before calling bind() and then bind the same port.

After downloading the tcp_client example for an ESP32 module, I connected the module to the router via Wi-Fi and performed a Ping test on the computer. Then the it shows high latency sometimes, what is the reason?

- When Wi-Fi is connected, Power Save mode will be turned on by default, which may cause high Ping delay. To solve this issue, you can turn off Power Save mode to reduce the delay by calling esp_wifi_set_ps(WIFI_PS_NONE) after esp_wifi_start().
I’m using ESP8266 release/v3.3 version of SDK to test the example/protocols/esp-mqtt/tcp example. Then during Wi-Fi configuration, the connection fails after configuring SSID, password and connecting to the default server. The log is as follows, what is the reason?

```
W (4211) MQTT_CLIENT: Connection refused, not authorized
I (4217) MQTT_CLIENT: Error MQTT Connected
I (4222) MQTT_CLIENT: Reconnect after 10000 ms
I (4228) MQTT_EXAMPLE: MQTT_EVENT_DISCONNECTED
I (19361) MQTT_CLIENT: Sending MQTT CONNECT message, type: 1, id: 0000
```

- When such error occurs, it is usually because the connected server is unavailable. Please use another available server for testing.

Using esp-idf release/v3.3 version of the SDK, is there an example for setting static IP for Ethernet?

- It can be set through the “tcpip_adapter_set_ip_info()” API, please refer to API description.
- Please refer to the example as follows:

```
/* Stop dhcp client */
tcpip_adapter_dhcpc_stop(TCPIP_ADAPTER_IF_STA);
/* static ip settings */
tcpip_adapter_ip_info_t sta_ip;
sta_ip.ip.addr = ipaddr_addr("192.168.1.102");
sta_ip.gw.addr = ipaddr_addr("192.168.1.1");
sta_ip.netmask.addr = ipaddr_addr("255.255.255.0");
tcpip_adapter_set_ip_info(TCPIP_ADAPTER_IF_STA, &sta_ip);
```

Does ESP32 have an LTE connection demo?

- Yes, please refer to the example/protocols/pppos_client demo in ESP-IDF v4.2 and later versions.

Will memory leak occur when ESP32 TCP repeatedly closes and rebuilds socket (IDF 3.3)?

- In ESP-IDF v3.3, every time a socket is created, a lock will be assigned, given that this internal socket array has not been assigned any lock before. This lock will not be reclaimed after the socket is released. Thus, next time the same socket array is allocated, the previous lock will be used again. That is to say, every time a new socket array is allocated and released, there will be one lock memory used. After all socket arrays being allocated, there will be no memory leak any more.
How to optimize memory when ESP32 uses mbedtls?

- You can enable dynamic buffer in menuconfig, the specific operation is menuconfig -> Component config -> mbedtls -> Using dynamic TX/RX buffer (key "Y" to enable).
- At the same time, you can enable the sub-options Free SSL peer certificate after its usage and Free certificate, key and DHM data after its usage in the Using dynamic TX/RX buffer in the previous step.

What is the default keep-alive value of the MQTT component in ESP-IDF?

- The default value is 120 s, which is defined by MQTT_KEEPALIVE_TICK in file mqtt_config.h.

Are there any limits on the maximum number of TCP client connection after the ESP32 additionally opens the TCP server?

- Yes. The number of simultaneously connected socket fd number for ESP32 is limited by LWIP_MAX_SOCKETS, which is 10 by default.

Does MQTT support automatic reconnection?

- The automatic reconnection of MQTT is controlled by the disable_auto_reconnect variable of struct esp_mqtt_client_config_t. The default value of disable_auto_reconnect is false, which means that automatic reconnection is enabled.
- The reconnection timeout value can be set using reconnect_timeout_ms.

What is the default MTU of LWIP for an ESP32?

- The default MTU of LWIP is 1500. This is a fixed value and it is not recommended to change it.

How to increase the DNS request time for ESP32?

- You can manually modify the #define DNS_MAX_RETRIES 4 in esp-idf/components/lwip/lwip/src/include/lwip/opt.h. For example, you can change the value of #define DNS_MAX_RETRIES to 10. In this way, the maximum time that DNS waits for a response from the server is 46 s (1+1+2+3+4+5+6+7+8+9).
How to use `esp_http_client` to send chunked data?

- Please use HTTP Stream by setting the `write_len` parameter of `esp_http_client_open()` to `-1`. Then the “Transfer-Encoding” will be set to “chunked” automatically please see `http_client_prepare_first_line()` in `esp_http_client.c`.

- The code snippet is listed below for your reference

```c
static void http_post_chunked_data()
{
    esp_http_client_config_t config = {
        .url = "http://httpbin.org/post",
        .method = HTTP_METHOD_POST,
        // This is NOT required. write_len < 0 will force → POST anyway
    };
    char buffer[MAX_HTTP_OUTPUT_BUFFER] = {0};
    esp_http_client_handle_t client = esp_http_client_init(&config);

    esp_err_t err = esp_http_client_open(client, -1); // write_len=-1 sets header → "Transfer-Encoding: chunked" and method to POST
    if (err != ESP_OK) {
        ESP_LOGE(TAG, "Failed to open HTTP connection: %s", esp_err_to_name(err));
        return;
    }

    // Post some data
    esp_http_client_write(client, "5", 1); // length
    esp_http_client_write(client, "\r\n", 2);
    esp_http_client_write(client, "Hello", 5); // data
    esp_http_client_write(client, "\r\n", 2);
    esp_http_client_write(client, "7", 1); // length
    esp_http_client_write(client, "\r\n", 2);
    esp_http_client_write(client, "World!", 7); // data
    esp_http_client_write(client, "\r\n", 2);
    esp_http_client_write(client, "0", 1); // end
    esp_http_client_write(client, "\r\n", 2);
    esp_http_client_write(client, "\r\n", 2);

    // After the POST is complete, you can examine the response as required using:
    int content_length = esp_http_client_fetch_headers(client);
    ESP_LOGI(TAG, "content_length: %d, status_code: %d", content_length, esp_http_ → client_get_status_code(client));

    int read_len = esp_http_client_read(client, buffer, 1024);
    ESP_LOGI(TAG, "receive %d data from server: %s", read_len, buffer);
    esp_http_client_close(client);
    esp_http_client_cleanup(client);
}
```
How to implement the certificate auto-download function?

CHIP: ESP32

- Please refer to aws certificate automatic download function.

After creating and closing TCP SOCKET several times, an error is reported as “Unable to create TCP socket: errno 23”. How to resolve such issue?

:CHIP: ESP8266 | ESP32 | ESP32-S2 | ESP32-C3 | ESP32-S3 :

- Reason: “errno 23” means open many open files in system. Closing a socket takes 2 MSL of time, which means sockets will not be closed immediately after calling the close interface. Due to this reason, open sockets are accumulated and exceeds the maximum connection number (the default is 10 in menuconfig, the maximum connection is 16) thus triggering this error.

- Solution: Set SO_LINGER via the setsockopt interface to adjust the TCP close time.

```c
linger link;
link.on_off = 1;
link.linger = 0;
setsockopt(m_sockConnect, SOL_SOCKET, SO_LINGER, (const char*)&link, sizeof(linger));
```

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_key.pem

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 `esptool.py summary` command from esptool.

How to enable flash encryption for ESP32?

- It can be enabled via `make menuconfig` or `idf.py menuconfig` \rightarrow Security features \rightarrow Enable flash encryption on boot (READ DOCS FIRST) configurations.
- Please refer to Flash encryption instructions.

After GPIO0 is pulled down, the ESP32 cannot enter download mode and prints “download mode is disable”. What is the reason?

- It means the chip’s UART Download mode has been disabled, you can check this via the `UART_DOWNLOAD_DIS` bit in efuse.
- Please note that after the Production mode of flash encryption is enabled, the UART Download mode will be disabled by default. For more information, please refer to UART ROM download mode.

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.

I'm using an ESP32 development board with the official PSRAM chip PSRAM64H embedded. But after replacing another type of PSRAM chip to PSRAM64H, it failed to recognize when I ran an ESP-IDF example and enabled the PSRAM configuration. What is the reason?

- If you need to change the PSRAM chip, please update configuration options in “menuconfig -> Component config -> ESP32-specific -> Support for external, SPI-connected RAM -> SPI RAM config -> Type of SPI RAM chip in use”.
- If you cannot find the corresponding type options of the new PSRAM chip you are about to use, please add the chip driver manually.
- It is recommended to use Espressif’s official ESP-PSRAM chip for ESP32 series.

What is the available size of RTC RAM in ESP8266 for users?

- The available RTC RAM in ESP8266 for users is ‘0x200’. Please see descriptions in esp8266.ld.

How to deal with the file with long filename when using the FAT filesystem

- You can change the configuration in menuconfig -> Component config -> FAT Filesystem support -> Long filename support by selecting the Long filename buffer in heap or Long filename buffer on stack option. Then you can modify the maximum length for a file name in Component config -> FAT Filesystem support -> Max long filename length.

How to enable exFAT?

CHIP: ESP32

- please modify #define FF_FS_EXFAT 0 as #define FF_FS_EXFAT 1, please refer to ffconf.h for details.
Will the configured Wi-Fi SSID and PASSWORD disappear after the ESP series development board is powered on again and need to be reconfigured?

- It will be stored in NVS by default and will not disappear due to power failure. You can also set it through `esp_wifi_set_storage()`, which can be divided into two situations:
  - If you want to save the Wi-Fi SSID and PASSWORD when powered off, you can store the Wi-Fi information in flash by calling `esp_wifi_set_storage(WIFI_STORAGE_FLASH)`.
  - If you want to achieve the operation of not saving the Wi-Fi SSID and PASSWORD when powered off, you can call `esp_wifi_set_storage(WIFI_STORAGE_RAM)` to store the Wi-Fi information in RAM.

Is there a limit to the number of partitions in the partition table of ESP32?

- The length of partition table is 0xC00 bytes (can store up to 95 partition table entries). Please refer to the description in partition table.

Can SPIFFS partition be encrypted?

**CHIP: ESP32, ESP32S2, ESP32S3, ESP32C3**

- No, there is no encryption scheme for SPIFFS. But, since SPIFFS is built on flash, this part of data can be encrypted via flash encryption.

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.

**When ESP8266 is in Deep sleep mode, can the data stored in RTC Memory work?**

- When ESP8266 is in Deep sleep mode, only the RTC timer continues to work. The data saved in the RTC Memory will not run, but can still be saved here. However, the data saved in RTC memory will lose after ESP8266 is powered off.

**What is the maximum length of the NVS Key for ESP32?**

- The maximum length of the NVS key for ESP32 is 15 characters, which cannot be changed. Please see the description of key-value pair.
- But you can use the value of nvs_set_str() to store data.

**Does the cJSON in ESP-IDF release/v4.2 support uint64_t data analysis?**

- No. The cJSON library has restrictions on parsing long integers, and the longest type is Double.

**Given that the GDB debugging function is working before the flash encryption is disabled, then why does the device keeps restarting during the GDB debugging after the flash encryption is enabled?**

- The JTAG debugging function will be disabled by default when flash encryption or secure boot is enabled. For more information, please refer to JTAG with Flash Encryption or Secure Boot.

**While using mobile’s hotspot for an ESP32 to download the OTA firmware, after a few seconds when turning of the hotspot and restarts ESP32, the program sticks in the OTA operation (the same situation for plugging and unplugging the wan port network cable when using a router), why?**

- This is a normal situation based on the protocol. When using the esp_https_ota component to run OTA, you can set the network timeout value (via http_config->timeout_ms) to 10 ~ 30 s (not too low) and enable http_config->keep_alive_enable to see if there are any errors in the link layer.
- If you are using an self-implemented OTA module, please set a timeout value via the select configuration or enable the TCP keep-alive mechanism to detect the link layer.
Why do different ESP32 modules have different flash erase time?

- This is caused by different type of flash models. Some module of flash don’t have a mechanism for passing empty blocks when erasing, so it takes longer time.

Which GPIOs can be used to wake up ESP32-C3 from Deep-Sleep mode?

- Only GPIO0 ~ GPIO5 in VDD3P3_RTC domain can be used to wake up ESP32-C3 from Deep-sleep mode. Please read Chapter 5.9.1 Power Supplies of GPIO Pins in ESP32-C3 Technical Reference Manual.

When using the ESP-WROOM-02D module with a battery for power supply, are there any risks in frequently formatted reading and writing flash as the battery is low (the module barely starts up)?

- In low power conditions, if the flash is frequently operated, it may accept error commands and then erase the flash at the wrong address. It is recommended to not to operate the flash when the power is off, and please ensure a stable power supply.

How to check the maximum stack size used by a thread for ESP32?

- You can call the UBaseType_t uxTaskGetStackHighWaterMark(TaskHandle_t xTask) function. This function will return the minimum remaining stack space after the task is started.

What is the meaning of the "SW_CPU_RESET" log when using ESP32?

- “SW_CPU_RESET” is the software reset log. For example, calling the “esp_restart()” API will print this log.

For ESP32 products, when testing NVS separately, I found it occupies a lot of memory. What is the reason?

- Please check the partition table settings. It is recommended to set a smaller NVS data partition in the partition table to test. The larger the NVS data partition setting, the more memory it will occupy.
How do I change the system time of a module?

CHIP: ESP32 | ESP32 | ESP32-C3
• You can use the c language `time()` interface to set the system time.

During the OTA upgrade process, an ESP_ERR_OTA_VALIDATE_FAILED error occurred after calling `esp_ota_end`, how to troubleshoot such issue?

CHIP: ESP32
• Generally it is caused by the error content in the downloaded firmware. You can dump out such content via `read_flash` in `esptool` from your module. Then use the Beyond Compare tool to compare the two bin files in hexadecimal to see which part of the bin file is downloaded incorrectly.

How does ESP8266-RTOS-SDK store data to RTC memory?

• The definition method of storing data in RTC memory is as follows:

After waking up from Deep-sleep mode, where does ESP8266 start boot?

• After ESP8266 wakes up from Deep-sleep mode, the device will boots up from `user_init`. Please refer to the description in `esp_deep_sleep()`.

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

```c
...
    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));
    }
    ...
}
```
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:

```c
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));

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);
```

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:

```c
// 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_ADDRLEASE_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)));
```

(continues on next page)
// 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.
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);`.

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.

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.
private static final int MIN_RSSI = -100;

/** Anything better than or equal to this will show the max bars. */
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: d0dae5426380f771b0e192d8cc501ce5308485e
  - 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

```java
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`.
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.
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.

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.

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.

When ESP32 serves as the transmitter, the first retransmission interval is normally 2.3 s by default. Then, the next interval is determined by Jacobson’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:

```c
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 station-Conf.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 frame Can 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


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:

```c
//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.

When using ESP32 to establish a hotspot, can I scan all APs and the occupied channels first, and then select the smallest and cleanest channel to establish my own AP?

• You can scan all APs and occupied channels before establishing a hotspot. Refer to the example "esp_wifi_scan_get_AP_records".
• It cannot be performed automatically. You need to customize the channel selection algorithm to implement such operation.

I’m scanning Wi-Fi on an ESP32 device using release/v3.3 version of ESP-IDF. When there are some identical SSIDs, same SSID names will show in the Wi-Fi list repeatedly. Is there an API to filter such repeated names?

• No, same SSID names cannot be filtered out since identical SSID names may not mean identical servers. Their BSSID may not be the same.

I’m using the master version of ESP8266-RTOS-SDK to open the WiFi Qos application to get EDCF support. How does ESP8266 decide which data packet should be allocated to the EDCF AC category?

• It can be determined by setting IPH_TOS_SET(iphdr, tos).

Using ESP-IDF release/v4.2 version of SDK, how to enable mDNS function in AP mode?

• Please enable “Component config -> LWIP -> Enable mDNS queries in resolving host name” in menuconfig.

Can Wi-Fi be used with ESP-NOW at the same time?

• Yes, but it should be noted that the channel of ESP-NOW must be the same as that of the connected AP.
Using ESP32, how to configure the maximum Wi-Fi transmission speed and stability without considering memory and power consumption?

- To configure the maximum Wi-Fi transmission speed and stability, please refer to How to improve Wi-Fi performance in ESP-IDF programming guide and set the relevant configuration parameters in menuconfig. The option path can be found by searching “/” in the menuconfig interface. The optimal configuration parameters need to be tested according to the actual environment.

In Wi-Fi SoftAP mode, how many Station devices can ESP8266 be connected at most?

- ESP8266 supports up to 8 Station device connections.

How to get CSI data when using ESP32 device in Station mode?

- CSI data can be obtained by calling ‘esp_wifi_set_csi_rx_cb()’. See description in API.
- See configuration steps in Wi-Fi CSI.

In AP + STA mode, after an ESP32 is connected to Wi-Fi, will the Wi-Fi connection be affected if I enable or disable its AP mode?

- After an ESP32 is connected to Wi-Fi in AP + STA dual mode, AP mode can be enabled or disabled at will without affecting Wi-Fi connection.

I’m using ESP-IDF release/v3.3 for ESP32 development, but only bluetooth function is needed, how to disable Wi-Fi function through software?

- Please call esp_wifi_stop() to disable the Wi-Fi function. For API description, please see esp_err_t esp_wifi_stop(void).
- If you need to reclaim the resources occupied by Wi-Fi, call esp_wifi_deinit(). For API description, please see esp_err_t esp_wifi_deinit(void).

In ESP-IDF, the esp_wifi_80211_tx() interface can only be used to send data packets, is there a corresponding function to receive packets?

- Please use callback function to received data packets as follows:

```c
Esp_wifi_set_promiscuous_rx_cb(wifi_sniffer_cb);
esp_wifi_set_promiscuous(true)
```
- The above data receive method is also used in another open-sourced project, please see esp-mdf.
Does ESP32 support seamless roaming between different APs with the same SSID?

• Currently not supported.

What are the reasons for the high failure rate of esptouch networking?

CHIP: ESP32, ESP32S2, ESP32S3, ESP32C3, ESP8266

• The same hotspot is connected too many people.
• The signal quality of the hotspot connected by cell phone is poor.
• The router does not forward multicast data.
• The router has enabled dual-band integration, and the phone is connected to the 5G frequency band.

How to optimize the IRAM when ESP32 uses Wi-Fi?

• You can disable WIFI_IRAM_OPT, WIFI_RX_IRAM_OPT and LWIP_IRAM_OPTIMIZATION in menu-config to optimize IRAM space, but this will degrade Wi-Fi performance.

How to test ESP32’s Wi-Fi transmission distance?

• You can use the iperf example and configure the ESP32 device to iperf UDP mode. Then, you can distance the device continuously to see at which point the Wi-Fi data transmission rate will drop to 0.

What is the maximum length of Wi-Fi MTU for an ESP32?

• The maximum Wi-Fi MTU length for ESP32 is 1500.

During the on-hook test for an ESP32 device, the following log shows. What does it mean?

log

```
```
- The value after `new` represents the current primary and secondary channel; the value after `old` represents the last primary and secondary channel; and the value after `ap` represents the primary and secondary channel of the current ESP32 AP, which will be 255 if softAP is not enabled; the value after `sta` represents primary and secondary channel of the current ESP32 sta; and `prof` is the channel of ESP32’s softAP stored in NVS.
- For the meaning of secondary channel values, please refer to `wifi_second_chan_t`.
- The above log indicates that router is switching between HT20 and HT40 minus. You can check the Wi-Fi bandwidth setting of the router.

### How to disable AP mode when ESP32 is in AP + STA mode?

- This can be done through the configuration of `esp_wifi_set_mode(wifi_mode_t mode);` function.
- Just call `esp_wifi_set_mode(WIFI_MODE_STA);`.

### After ESP32 used the Wi-Fi function, are all ADC2 channels unavailable?

- When an ESP32 device is using Wi-Fi function, the ADC2 pins that are not occupied by Wi-Fi can be used as normal GPIOs. You can refer to the official ADC Description.

### How do I set the country code for a Wi-Fi module?

**CHIP: ESP8266 | ESP32 | ESP32 | ESP32-C3**

- Please call `esp_wifi_set_country` to set the country code.

### When using ESP32 as a SoftAP and have it connected to an Iphone, a warning prompts as “low security WPA/WPA2(TKIP) is not secure. If this is your wireless LAN, please configure the router to use WPA2(AES) or WPA3 security type”, how to solve it?

**IDF: release/v4.0 and above**

- You can refer to the following code snippet:

```c
wifi_config_t wifi_config = {
    .ap = {
        .ssid = EXAMPLE_ESP_WIFI_SSID,
        .ssid_len = strlen(EXAMPLE_ESP_WIFI_SSID),
        .channel = EXAMPLE_ESP_WIFI_CHANNEL,
        .password = EXAMPLE_ESP_WIFI_PASS,
        .max_connection = EXAMPLE_MAX_STA_CONN,
        .authmode = WIFI_AUTH_WPA2_PSK,
        .pairwise_cipher = WIFI_CIPHER_TYPE_CCMP
    }
};
```
• WIFI_AUTH_WPA2_PSK is AES, also called CCMP. WIFI_AUTH_WPA_PSK is TKIP. WIFI_AUTH_WPA_WPA2_PSK is TKIP+CCMP.

Since ESP32’s Wi-Fi module only supports 2.4 GHz of bandwidth, can Wi-Fi networking succeed when using a multi-frequency router with both 2.4 GHz and 5 GHz of bandwidth?

• Please set your router to multi-frequency mode (can support 2.4 GHz and 5 GHz for one Wi-Fi account), and the ESP32 device can connect to Wi-Fi normally.

How to obtain the RSSI of the station connected when ESP32 is used in AP mode?

• You can call API esp_wifi_ap_get_sta_list, please refer to the following code snippet:

```
{   wifi_sta_list_t wifi_sta_list;
    esp_wifi_ap_get_sta_list(&wifi_sta_list);
    for (int i = 0; i < wifi_sta_list.num; i++) {
        printf("mac address: %02x:%02x:%02x:%02x:%02x:%02x\t rssi:%d\n",wifi_sta_list.sta[i].mac[0], wifi_sta_list.sta[i].mac[1],wifi_sta_list.sta[i].mac[2], wifi_sta_list.sta[i].mac[3],wifi_sta_list.sta[i].mac[4],wifi_list.sta[i].mac[5], wifi_sta_list.sta[i].rssi);
    }
}
```

• The RSSI obtained by esp_wifi_ap_get_sta_list is the average value over a period of time, not real-time RSSI. The previous RSSI has a weight of 13, and the new RSSI has a weight of 3. The RSSI is updated when it is or larger than 100ms, the old rssi_arg is used when updating as: rssi_avg = rssi_avg*13/16 + new_rssi * 3/16.

Does ESP32 support FTM(Fine Timing Measurement)?

• No, it doesn’t. FTM needs hardware support, but ESP32 doesn’t have it.
• ESP32-S2 and ESP32-C3 can support FTM in hardware.
• ESP-IDF can support FTM from v4.3-beta1.
• For more information and examples of FTM, please refer to FTM.
When ESP32 is in STA+AP mode, how to specify whether using STA or AP interface to send data?

**Background:**

The default network segment of ESP32 as AP is 192.168.4.x, and the network segment of the router to which ESP32 as STA is connected is also 192.168.4.x. The PC connects to the same router and creates a tcp server. In this case, the tcp connection between ESP32 as tcp client and PC as tcp server cannot be established successfully.

**Solutions:**

- It is possible for ESP32 to specify whether to use STA or AP interface for data transmission. Please see example `tcp_client_multi_net`, in which both ethernet and station interface are used and each can be specified for data transmission.

- There are two ways to bind socket to an interface:
  - use netif name (use socket option SO_BINDTODEVICE)
  - use netif local IP address (get IP address of an interface via `esp_netif_get_ip_info()`, then call bind())

**Note:**

- The tcp connection between ESP32 and PC can be established when an ESP32 is bound to the STA interface, while the connection cannot be established when it is bound to the AP interface.

- By default, the tcp connection between ESP32 and mobile phone can be established (the mobile phone as a station is connected to ESP32).

---

**ESP8266 wpa2_enterprise How to enable Wi-Fi debugging function?**

- Open menuconfig via `idf.py menuconfig` and configure the following parameters:

```
menuconfig==>Component config ==>Wi-Fi ==>  
[*]Enable WiFi debug log ==>The DEBUG level is enabled (Verbose) 
[*]WiFi debug log submodule 
[*] scan 
[*] NET80211 
[*] wpa 
[*] wpa2 Enterprise 

menuconfig==>Component config ==>Supplicant ==>[*] Print debug messages from WPA
←Supplicant
```

---

116 Chapter 1. Instruction
Is there a standard for the number of Wi-Fi signal frames?

CHIP: ESP8266 | ESP32 | ESP32 | ESP32-C3

• There is no such standard for now. You can do the calculation by yourself based on the received RSSI. For example, if the received RSSI range is [0,-96], and the required signal strength is 5, then [0~-20] is the full signal, and so on.

What is the current progress of WFA bugs fixing?

CHIP: ESP32 | ESP32-S2 | ESP32-C3 | ESP8266


When Wi-Fi connection failed, what does the error code mean?

CHIP: ESP32

• Any error occurred during the Wi-Fi connection will cause it coming to init status, and there will be a hexadecimal number in the log, e.g., wifi:state, auth-> init(200). The first two digits indicate error reasons while the last two digits indicate the type code of the received or transmitted management frame. Common frame type codes are 00 (received nothing, timeout), A0 (disassoc), B0 (auth) and C0 (deauth).

• Error reasons indicated by the first two digits can be found in Wi-Fi Reason Code. The last two digits can be checked in frame management code directly.

When using ESP32’s Release/v3.3 of SDK to download the Station example, the device cannot be connected to an unencrypted Wi-Fi. What is the reason?

• In the example, it is by default to connect to an encrypted AP as:

```c
.threshold.authmode = WIFI_AUTH_WPA2_PSK,
```

• If you need connect to an unencrypted AP, please set the following parameter to 0:

```c
.threshold.authmode = 0,
```

• For AP mode selection instructions, please refer to esp_wifi_types.
What is the maximum speed of Wi-Fi communication of ESP32-S2 chip?

- The theoretical maximum speed of ESP32-S2 Wi-Fi communication is 150 Mbps.

What does such log mean:  I (81447377) wifi:new:<7,0>, old:<7,2>, ap:<255,255>, sta:<7,0>, prof:1?

:CHIP: ESP32 :

- new represents the current primary and secondary channel; old represents the last primary and secondary channel; ap represents the current primary and secondary channel of ESP32 AP; <255, 255> means SoftAP is disabled; sta represents the current primary and secondary channel of ESP32 STA; prof represents the channel of SP32 SoftAP stored in NVS.

Does ESP32 Wi-Fi support roam function between different APs with the same SSID?

- No.

Does ESP modules support EAP-FAST?

:CHIP: ESP32 | ESP32-S2 | ESP32-C3 :

- Yes, please refer to wifi_eap_fast demo.

Does ESP modules support the WiFi NAN (Neighbor Awareness Networking) protocol?

CHIP: ESP8266 | ESP32 | ESP32-C3 | ESP32-S2 | ESP32-S3

- No.

Does ESP8266 support EDCF (AC) scheme?

- The master version of ESP8266-RTOS-SDK supports EDCF (AC) applications, but no application examples are provided for now. You can enable WiFi QoS configuration in menuconfig -> Component config -> Wi-Fi to get support.
When using ESP32 with release/v3.3 version of ESP-IDF. When configuring the router, is there an API to directly tell that the entered password is wrong?

- There is no such API. According to the Wi-Fi protocol standard, when the password is wrong, the router will not clearly tell the Station that the 4-way handshake is due to the password error. Under normal circumstances, the password is obtained in 4 packets (1/4 frame, 2/4 frame, 3/4 frame, 4/4 frame). When the password is correct, the AP will send 3/4 frames, but when the password is wrong, the AP will not send 3/4 frame but send 1/4 frame instead. However, when the AP sends 3/4 frame which is lost in the air for some reason, the AP will also re-send 1/4 frame. Therefore, for Station, it is impossible to accurately distinguish between these two situations. In the end, it will report a 204 error or a 14 error.

- Please refer to Wi-Fi reason code.

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.
Do ESP modules support Thread?

CHIP: ESP32 | ESP32-C3 | ESP32-S3

• No. Currently, only ESP32-H2 supports Thread.

Does ESP modules support WAPI (Wireless LAN Authentication and Privacy Infrastructure)?

• Yes.

Development board

[]

How long does it take for the ESP-WROOM-02D module to restart after the reset signal?

• It will restart when the input level is lower than 0.6 V for more than 200 s.

According to the schematic of ESP32-LyraT-Mini, the analog output of the ES8311 codec chip is connected to the input of the ES7243 ADC chip. What is the purpose of this?

• To use the ES7243 ADC chip to obtain the necessary input reference signal for AEC.

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 \texttt{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.

How to configure the RMII synchronization clock for Ethernet of the ESP32?

- Please download the “esp-idf/examples/ethernet/basic” example for testing.
- The IP101 PHY chip will experience network instability when GPIO0 outputs CLK. Therefore, it is recommended to connect a 50MHz crystal oscillator to the PHY and GPIO0 as input.
- Due to the special character of GPIO0, it is necessary to configure the IO to control the enable pin of the PHY.
- Please read Configure MAC and PHY.
- You can see SCH_ESP32-ETHERNET-KIT schematic design for reference.

How to hardware reset ESP8266? Is the hardware reset signal low level or high level active? What are the conditions for the reset?

- The Pin32 EXT_RSTB of ESP8266 is the reset pin. This pin has an internal pull-up resistor, active in low level. To prevent the restart caused by external interference, it is recommended that the EXT_RSTB cabling be as short as possible and that an RC circuit be added to the EXT_RSTB pin.
- ESP8266’s CHIP_EN pin can also be used as a hardware reset pin. When using the CHIP_EN pin as a reset pin, the reset signal is low level effective. The reset condition is: when the input level is lower than 0.6 V and lasts more than 200 s, then the ESP8266 will be reset and restart. It is recommended to use the CHIP_EN pin for chip reset. Please refer to Section “1.4.2.2 Reset” in ESP8266 Hardware Design Guide for more information.
What does the term NC mean in Espressif schematics?

- NC is the acronym of No Component. If you see a pull-up resistor is marked NC as shown in the figure below, it indicates that the component is not installed.

![NC Example](image.png)

How to use multiple antennas in ESP32-S2?

- It is similar to ESP32, you can refer to ESP32-WROOM-DA.
- For detailed operation instructions, please refer to ESP-IDF Programming Guide.
- You can add an RF switch to select a working antenna.

Does ESP32-C3F SPI CS0 pin need an external 10 K pull-up resistor?

**CHIP: ESP32-C3F**

- No.

Is there any hardware design reference for ESP-Skainet Speech recognition?

- Please refer to `ESP32-Korvo V1.1 Hardware Reference Design <https://github.com/espressif/esp-skainet/blob/master/docs/en/hw-reference/esp32/user-guide-esp32-korvo-v1.1.md#2%E7%A1%AC%E4%BB%B6%E5%8F%82%E8%80%83>`_.


Is it necessary to connect a 32 KHz RTC crystal?

CHIP: ESP32 | ESP32-C3 | ESP32-S3

- The external 32KHz crystal is mainly used for BLE light sleep timing, so in applications where BLE light sleep is not used, there is no need to do so.

Does ESP8266 support 32 MHz crystal frequency?

- No, ESP8266 supports 26 MHz and 40 MHz crystal frequencies, and 26 MHz is recommended.

RF related

[]

If an ESP32 module is running under a 2.8 V supply, is there any degradation in its RF performance?

Yes, its RF performance may become unstable. It is recommended that the voltage be supplied in accordance with the suggested operating voltage range specified in the Module's Datasheet.

What are the modulation methods supported by Espressif’s chips?

- ESP8266 supports: BPSK/QPSK/16QAM/64QAM/DBPSK/DQPSK/CCK.
- ESP32 supports: BPSK/QPSK/16QAM/64QAM/DBPSK/DQPSK/CCK/GFSK /4-DQPSK 8-DPSK.
- ESP32-S2 supports: BPSK/QPSK/16QAM/64QAM/DBPSK/DQPSK/CCK.

How can I get the RF related information (e.g., antenna specification, antenna pattern, etc.) for certification?

For such information, please contact Sales for help.

Why does ESP32 automatically reduce the transmit power when running at a high temperature of 80 °C when using the RF Test Tool?

- The temperature compensation function on testing firmware with a fixed frequency is disabled by default. Therefore, when the temperature is high, the power will be lower. If you need to enable the temperature compensation, please send txpwr_track_en 1 1 0 to ESP32 through the default log serial port.
How to improve the distance and strength of Wi-Fi signals for ESP32-WROVER-E? (Application scenario: Wi-Fi probe)

- In terms of software, you can either set the maximum transmit power by API `esp_wifi_set_max_tx_power()`, or set that via menuconfig: `Component config -> PHY -> Max Wi-Fi TX power(dBm)` (the default transmit power is 20 dBm).

- If the transmit power has been set to the maximum, you can improve the efficiency the antenna and receiving device.
  - You can consider adjusting the placement direction of the module so that the stronger radiation direction of the antenna points to the receiving device to make the farthest radiation distance.
  - Make sure there is no metal or blocking object near the antenna of the module, no PCB on the back of the antenna, and the Wi-Fi signal is not interfered by other signals of the whole machine.
  - If the PCB antenna is not effective, you can use the IE series module with an external IPEX antenna with higher directional gain.
  - The receiving device can also increase the antenna radiation efficiency.

How to write phy_init data to Flash?

:CHIP: ESP32 :

- You can write via the power limit tool. Please download the `ESP_RF_TEST Tool`, unzip the package and open the `EspRFTestTool_v2.6_Manual.exe` file, then click `help ---> Tool help ---> PowerLimitTool help` for detailed operations.

Process and ESD

[]

Production test

[]

How to get the production test tool?

:CHIP: ESP32 | ESP8266 :

- Please download `Production Test Tool`.

Espressif Q & A.
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.
What is the power consumption of ESP8266 when the CHIP_PU pin is low?

- CHIP_PU pin is the module EN pin. When the pin is set to low level, the power consumption of the chip is about 0.5 μA.
- In ESP8266 datasheet Table 3-4, the power consumption mode is off, which means the CHIP_PU is pulled down and the chip is disabled.

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:

```bash
$ 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 Guidelines

Please add new Q&A items and do updates according to the guidelines as follows:

General guideline:

- If you are going to add a new Q&A item, always remember to add a separate line after the previous one as “———-”.

For questions:

- Illustrate questions clearly, for example:
  - When flashing firmware to ESP32-S2, an error occurred as “A fatal error occurred: Invalid head of packet (0x50)? (NOT Clear)
  - When flashing firmware to ESP32-S2, an error occurred as “A fatal error occurred: Invalid head of packet (0x50)”. How to resolve such issue? (Preferable)
- Do not make the question too long. If this is the case, extract the main question as the title and describe the background and details below.

For answers:

- If there will be code in the text, use code box to separate it with the main text.
- If an answer only includes one sentence, there is no need to write a list, just use a regular paragraph.
- Use lists to separate items or to enumerate sequential items:
  - Use numbered lists for items that are in a required order (such as step-by-step instructions) or for items that are referred to by item number.
  - Use bulleted lists for items that are in no required order.
- Provide introductory phrase before a list to indicate the meaning or purpose of the list, and place a colon “:” at the end of it.
- If two items are alternatives, use a bullet list (not numbered list) and indicate their relationship in the introductory phrase.
• Always add two spaces before an listed item or paragraph in the answer.
• When a separate notice or explanatory paragraph follows a item, indent that separate material to the text of the parent list item.
• Follow list punctuation rules described in Espressif Manual of Style, Section Punctuation in Lists.

For additional guidance regarding list please refer to Bulleted and Numbered Lists. Please see the example template for text and figures below.

**Q&A 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
```

(continues on next page)
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

decactivate

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.