**Over The Air Updates**

## **OTA Data Partition:**

## An OTA data partition (type data, subtype ota) must be included in the [Partition Tables](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/partition-tables.html) of any project which uses the OTA functions. For factory boot settings, the OTA data partition should contain no data (all bytes erased to 0xFF). In this case, the ESP-IDF software bootloader will boot the factory app if it is present in the partition table. If no factory app is included in the partition table, the first available OTA slot (usually ota\_0) is booted. After the first OTA update, the OTA data partition is updated to specify which OTA app slot partition should be booted next. The OTA data partition is two flash sectors (0x2000 bytes) in size, to prevent problems if there is a power failure while it is being written. Sectors are independently erased and written with matching data, and if they disagree a counter field is used to determine which sector was written more recently.

## **App Rollback:**

The main purpose of the application rollback is to keep the device working after the update.

* The application works fine, [**esp\_ota\_mark\_app\_valid\_cancel\_rollback()**](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html#_CPPv438esp_ota_mark_app_valid_cancel_rollbackv) marks the running application with the state ESP\_OTA\_IMG\_VALID. There are no restrictions on booting this application.
* The application has critical errors and further work is not possible, a rollback to the previous application is required, [**esp\_ota\_mark\_app\_invalid\_rollback\_and\_reboot()**](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html#_CPPv444esp_ota_mark_app_invalid_rollback_and_rebootv) marks the running application with the state ESP\_OTA\_IMG\_INVALID and reset. This application will not be selected by the bootloader for boot and will boot the previously working application.
* If the [CONFIG\_BOOTLOADER\_APP\_ROLLBACK\_ENABLE](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/kconfig.html#config-bootloader-app-rollback-enable) option is set, and a reset occurs without calling either function then the application is rolled back.

### **App OTA State:**

States control the process of selecting a boot app:

| **States** | **Restriction of selecting a boot app in bootloader** |
| --- | --- |
| ESP\_OTA\_IG\_ VALID | None restriction. Will be selected. |
| ESP\_OTA\_IMG\_UNDEFINED | None restriction. Will be selected. |
| ESP\_OTA\_IMG\_INVALID | Will not be selected. |
| ESP\_OTA\_IMG\_ABORTED | Will not be selected. |
| ESP\_OTA\_IMG\_NEW | If [CONFIG\_BOOTLOADER\_APP\_ROLLBACK\_ENABLE](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/kconfig.html#config-bootloader-app-rollback-enable) option is set it will be selected only once. In bootloader the state immediately changes to ESP\_OTA\_IMG\_PENDING\_VERIFY. |
| ESP\_OTA\_IMG\_PENDING\_VERIFY | If [CONFIG\_BOOTLOADER\_APP\_ROLLBACK\_ENABLE](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/kconfig.html#config-bootloader-app-rollback-enable) option is set it will not be selected, and the state will change to ESP\_OTA\_IMG\_ABORTED. |

If [CONFIG\_BOOTLOADER\_APP\_ROLLBACK\_ENABLE](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/kconfig.html#config-bootloader-app-rollback-enable) option is not enabled (by default), then the use of the following functions [**esp\_ota\_mark\_app\_valid\_cancel\_rollback()**](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html#_CPPv438esp_ota_mark_app_valid_cancel_rollbackv) and [**esp\_ota\_mark\_app\_invalid\_rollback\_and\_reboot()**](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html#_CPPv444esp_ota_mark_app_invalid_rollback_and_rebootv) are optional, and ESP\_OTA\_IMG\_NEW and ESP\_OTA\_IMG\_PENDING\_VERIFY states are not used.

An option in Kconfig [CONFIG\_BOOTLOADER\_APP\_ROLLBACK\_ENABLE](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/kconfig.html#config-bootloader-app-rollback-enable) allows you to track the first boot of a new application. In this case, the application must confirm its operability by calling [**esp\_ota\_mark\_app\_valid\_cancel\_rollback()**](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html#_CPPv438esp_ota_mark_app_valid_cancel_rollbackv) function, otherwise the application will be rolled back upon reboot. It allows you to control the operability of the application during the boot phase. Thus, a new application has only one attempt to boot successfully.

### **Rollback Process:**

The description of the rollback process when [CONFIG\_BOOTLOADER\_APP\_ROLLBACK\_ENABLE](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/kconfig.html#config-bootloader-app-rollback-enable) option is enabled:

* The new application is successfully downloaded and [**esp\_ota\_set\_boot\_partition()**](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html#_CPPv426esp_ota_set_boot_partitionPK15esp_partition_t) function makes this partition bootable and sets the state ESP\_OTA\_IMG\_NEW. This state means that the application is new and should be monitored for its first boot.
* Reboot [**esp\_restart()**](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/misc_system_api.html#_CPPv411esp_restartv).
* The bootloader checks for the ESP\_OTA\_IMG\_PENDING\_VERIFY state if it is set, then it will be written to ESP\_OTA\_IMG\_ABORTED.
* The bootloader selects a new application to boot so that the state is not set as ESP\_OTA\_IMG\_INVALID or ESP\_OTA\_IMG\_ABORTED.
* The bootloader checks the selected application for ESP\_OTA\_IMG\_NEW state if it is set, then it will be written to ESP\_OTA\_IMG\_PENDING\_VERIFY. This state means that the application requires confirmation of its operability, if this does not happen and a reboot occurs, this state will be overwritten to ESP\_OTA\_IMG\_ABORTED (see above) and this application will no longer be able to start, i.e., there will be a rollback to the previous working application.
* A new application has started and should make a self-test.
* If the self-test has completed successfully, then you must call the function [**esp\_ota\_mark\_app\_valid\_cancel\_rollback()**](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html#_CPPv438esp_ota_mark_app_valid_cancel_rollbackv) because the application is awaiting confirmation of operability (ESP\_OTA\_IMG\_PENDING\_VERIFY state).
* If the self-test fails, then call [**esp\_ota\_mark\_app\_invalid\_rollback\_and\_reboot()**](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html#_CPPv444esp_ota_mark_app_invalid_rollback_and_rebootv) function to roll back to the previous working application, while the invalid application is set ESP\_OTA\_IMG\_INVALID state.
* If the application has not been confirmed, the state remains ESP\_OTA\_IMG\_PENDING\_VERIFY, and the next boot it will be changed to ESP\_OTA\_IMG\_ABORTED, which prevents re-boot of this application. There will be a rollback to the previous working application.

### **Unexpected Reset:**

If a power loss or an unexpected crash occurs at the time of the first boot of a new application, it will roll back the application.

Recommendation: Perform the self-test procedure as quickly as possible, to prevent rollback due to power loss.

Only OTA partitions can be rolled back. Factory partition is not rolled back.

### Booting Invalid/aborted Apps[ℑ](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html#booting-invalid-aborted-apps)

Booting an application which was previously set to ESP\_OTA\_IMG\_INVALID or ESP\_OTA\_IMG\_ABORTED is possible:

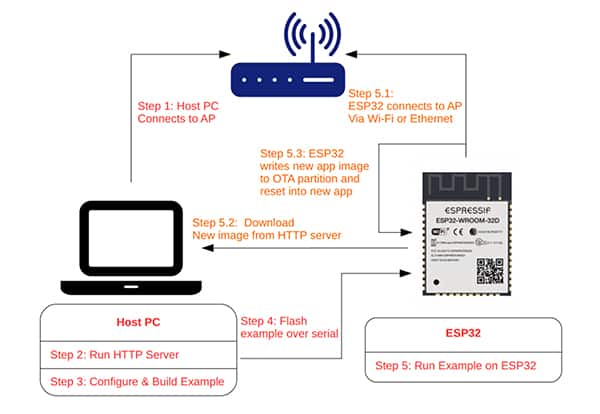
* Get the last invalid application partition [**esp\_ota\_get\_last\_invalid\_partition()**](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html#_CPPv434esp_ota_get_last_invalid_partitionv).
* Pass the received partition to [**esp\_ota\_set\_boot\_partition()**](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html#_CPPv426esp_ota_set_boot_partitionPK15esp_partition_t), this will update the otadata.
* Restart [**esp\_restart()**](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/misc_system_api.html#_CPPv411esp_restartv). The bootloader will boot the specified application.

To determine if self-tests should be run during startup of an application, call the [**esp\_ota\_get\_state\_partition()**](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html#_CPPv427esp_ota_get_state_partitionPK15esp_partition_tP20esp_ota_img_states_t) function. If result is ESP\_OTA\_IMG\_PENDING\_VERIFY then self-testing and subsequent confirmation of operability is required.

## **Anti-rollback**[**:**](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html#anti-rollback)

Anti-rollback prevents rollback to application with security version lower than one programmed in eFuse of chip.

This function works if set [CONFIG\_BOOTLOADER\_APP\_ANTI\_ROLLBACK](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/kconfig.html#config-bootloader-app-anti-rollback) option. In the bootloader, when selecting a bootable application, an additional security version check is added which is on the chip and in the application image.



* [Over-the-Air Firmware Updates Using the ESP32 MCU | DigiKey](https://www.digikey.in/en/articles/how-to-perform-ota-updates-using-esp32-microcontroller-and-esp-idf)
* [Over The Air Updates (OTA) - ESP32 - — ESP-IDF Programming Guide latest documentation (espressif.com)](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/ota.html)