Flash Download Tool

User Guide

Related Product

ESP32 series

ESP8266 series

ESP32-S2 series

ESP32-C3 series

ESP32-S3 series

ESP32-C2 series

ESP32-C6 series

ESP32-H2 series



About This Guide

This document describes how to download and configure firmware on Espressif modules, using Espressif's Flash Download Tool. It also lists some frequently asked questions and answers. This document is applicable to Flash Download Tool v3.9.6.

Release notes

Date	Version	Release notes
2018.08	v1.0	Initial release.
2019.03	v1.1	 Updated Sections 3.2.2.5, 3.5, 5.1, and Appendix A; Added Section 4.1.3; Removed Section 5.6.
2020.04	v1.2	Update a typo in Section 4.3;Update the description of the note in Section 4.3.
2020.07	v1.3	Added feedback links.
2021.12	v1.4	 Added description of mass production mode. Updated description of encryption configuration. Updated ways of selecting chips. Deleted RFConfig Chapter. Simplified document structure. Deleted description of flash size configuration. Deleted description of spi auto set configuration. Deleted description of GPIO configuration interface. Support download over USB. Simplified document description. Updated document format.
2023.05	v1.5	 Added more encryption configuration options and the descriptions.
2023.12	v1.6	 Added descriptions of LoadMode for ESP32-C2.
2024.04	v1.7	Updated some configuration items, including [SECURE BOOT], [SECURE OTHER CONFIG], [FLASH ENCRYPTION KEYS LOCAL SAVE], [ESP32* EFUSE BIT CONFIG]

Documentation Change Notification

Espressif provides email notifications to keep customers updated on changes to technical documentation. Please subscribe at https://www.espressif.com/en/subscribe.

Certification

Download certificates for Espressif products from https://www.espressif.com/en/certificates.

Table of Contents

1.	Prepa	aration	1	
2.	Tool Overview			
	2.1.	User Interface	2	
	2.2.	SPIDownload Tab	2	
	2.3.	HSPIDownload Tab	5	
	2.4.	FactoryMultiDownload Tab	5	
3.	Download Example			
	3.1.	Regular Download Example	6	
	3.2.	Enable Encryption for Firmware Downloading	7	
4.	FAQ.		11	
	4.1.	COM Related Errors	11	
	4.2.	Synchronization Related Errors	11	
	4.3.	eFuse Related Errors	11	
	4.4.	Download Related Errors	12	
	4.5.	Operation Related Errors	12	
aaA	endix	A. Contents of the Flash Download Tool Folder	13	



1. Preparation

The software and hardware resources required for downloading firmware to flash are listed below.

- Hardware:
 - o 1 x module to which firmware is downloaded
 - o 1 x PC (Windows 7 [64 bits], Windows 10)
- Software:

Download Tool: <u>Flash Download Tool</u> (For the detailed structure of this tool, please refer to Appendix A.)



2. Tool Overview

2.1. User Interface

Open the Flash Download Tool package, double-click the .exe file to enter the main interface of the tool, as shown in the figure below:

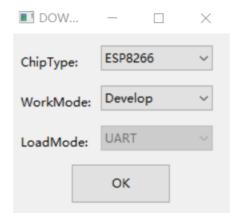


Figure 2-1. Flash Download Tool Main Interface

ChipType: select the chip type according to what product you use.

WorkMode: work mode of the tool. Below are the differences between the two modes supported currently – Develop and Factory modes.

- Developer mode uses the absolute path of the firmware and only allows flashing firmware to one chip at a time.
- Factory mode uses a relative path. It is recommended to place the firmware to flash in the bin directory of this tool package. It will be automatically saved locally when closed after configuration.
- Selecting Factory mode leads you to a locked interface in order to prevent misoperation by your mouse. Please click the LockSettings button to enable editing.

LoadMode: Download interface. Currently, ESP8266, ESP8285, and ESP32, ESP32-C2 only support UART, and other chip types support both UART and USB.

2.2. SPIDownload Tab

Here is the configuration descriptions.

• Download Path Config

You can configure the firmware loading path and downloading address (in hexadecimal format), such as 0x1000.

- SPI Flash Config
 - o SPI SPEED: SPI boot rate.
 - o SPI MODE: SPI boot mode.



- o **DETECTED INFO**: flash & crystal oscillator information that are detected automatically.
- DoNotChgBin: If it is enabled, the tool flashes the original content of the bin file. If not enabled, the tool updates the firmware according to the SPI SPEED, SPI MODE configuration on the interface before flashing.
- CombineBin button: combines all the selected firmware in Download Path Config into one firmware. If DoNotChgBin is enabled, combine the original firmwares. If DoNotChgBin is not enabled, combine them according to the SPI SPEED and SPI MODE configuration. Any unused areas between firmware files will be filled with 0xff. The combined firmware will be saved as ./combine/target.bin. Each click of this button will overwrite the previous firmware.



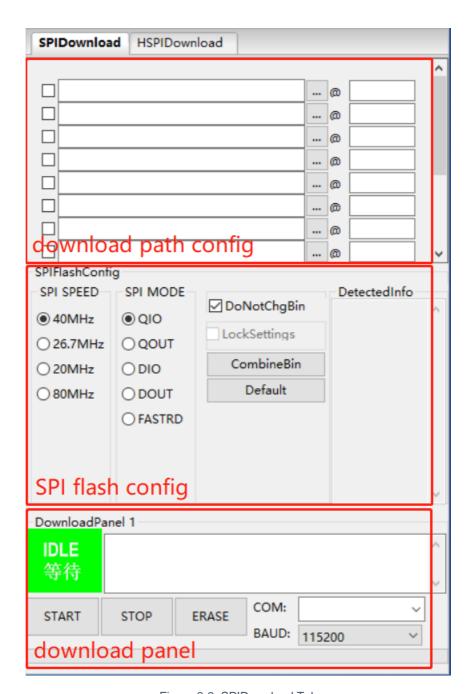


Figure 2-2. SPIDownload Tab

o **Default** button: restores the SPI configuration to the default values.

• Download Panel

o START: start downloading

o STOP: stop downloading

o ERASE: erase the entire flash

o COM: serial port used for downloading

o BAUD: baud rate



2.3. HSPIDownload Tab

The **SPIDownload** tab is needed only for the ESP8266 series of chips that connect to external flash via HSPI. It has the same interface as the **HSPIDownload** tab, so please refer to Section 2.2 SPIDownload Tab for interface description.

2.4. FactoryMultiDownload Tab

- Factory mode uses the relative path. By default, the tool loads the firmware from
 the bin folder of the tool directory. Whereas, Develop mode uses the absolute
 path. The advantage of the Factory mode is that as long as the firmware to flash
 remains in the bin folder of the tool directory, path problems will not occur when
 the tool package is copied to other factory computers.
- In Factory mode, the tool enables LockSettings by default. When LockSettings is
 enabled, firmware download path config and SPI flash config cannot be
 configured. This is to prevent production line workers from accidentally clicking
 and causing errors. (When factory managers need to configure these settings,
 they can click LockSettings to unlock.)

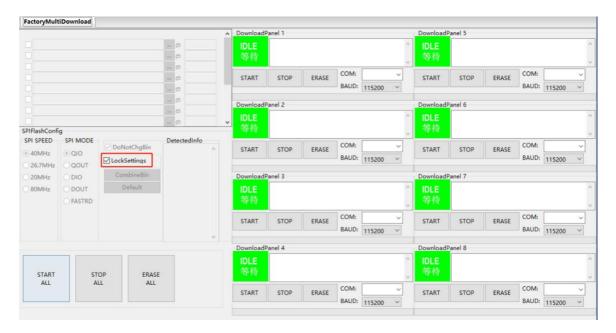


Figure 2-3. FactoryMultiDownload Tab

The download path config and SPI flash config section on the FactoryMultiDownload are basically the same as those on the SPIDownload tab. Please refer to Section 2.2 SPIDownload Tab for descriptions. Do not forget to configure the serial port number and baud rate of each download panel.



3. Download Example

This chapter takes the ESP32 series as an example to demonstrate how to perform both regular and encrypted download operations. At present, all chips series support regular download, and only ESP32 supports encrypted download. Other chip series will support encrypted download later.

3.1. Regular Download Example

- 1. Set the device to download mode:
 - o ESP32, ESP32-S2, ESP32-S3, ESP8266: pull GPIO0 low to enter the download mode.
 - o ESP32-C3: pull GPIO9 low and GPIO8 high to enter the downloading mode.
- 2. Open the download tool, set **ChipType** to ESP32, **WorkMode** to Develop, and **LoadMode** to UART as shown in the figure below. Then, click **OK**

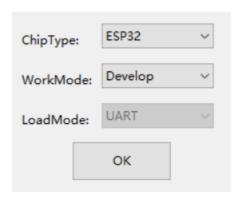


Figure 3-1. Selecting Device — ESP32 Download Tool

- 3. In the appeared download page, enter the path to the bin file and the address where it should be downloaded, check the box before the path, and select SPI SPEED, SPI MODE, COM, and BAUD according to your actual needs.
- 4. Click **START** to start downloading. During the download process, the tool will read the flash information and the chip's MAC address.
- 5. After the download is complete, the tool interface is shown in Figure 3-2.



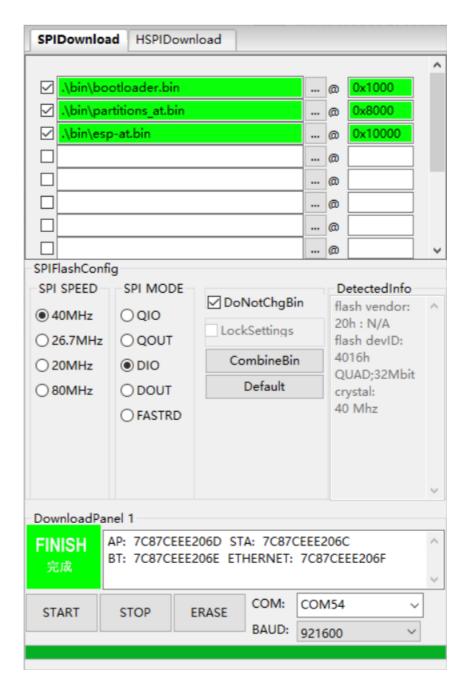


Figure 3-2. Download Completed

3.2. Enable Encryption for Firmware Downloading

The encrypted firmware downloading process is as follows:

- The Flash Download Tool downloads the plaintext firmware to the chip.
- The chip uses the key in its eFuse to encrypt the firmware and write it to the flash.
- If there is no such key in the eFuse, the tool will automatically generate a random one and flash it to eFuse. If there is, the tool skips the key generation and flashing process.

To configure the encryption function, follow the steps below:

• Open the configuration file ./configure/esp32/security.conf. If there is no such file, for example, when you open the tool for the first time, restart the tool.



• Update the configuration options as needed.

Below are the configuration options. The equal sign is followed by the default value of the option. "True" means enabling the option; "False" means disabling it.

• [SECURE BOOT]

Secure boot related configurations:

- secure_boot_en = False (Configures whether to enable secure boot)
- secure_boot_version = 1 (Selects secure boot version, ESP32 only)
- public_key_digest_path = .\secure\public_key_digest.bin (Path to the public key digest file. This file is generated using the command espsecure digest_sbv2_public_key -k pem.pem -o public_key_digest.bin. The .pem file is the private key file specified during compilation.)
- public_key_digest_block_index = 0 (Index of the eFuse block where the public key digest file is stored. Default: 0.)

• [FLASH ENCRYPTION]

Flash encryption related configurations:

- flash_encryption_en = False (Configures whether to enable flash encryption)
- reserved_burn_times = 3 (Configures how many times [3 in this case] are reserved for the flashing operation)
- [ESP32-C* and ESP 32-S* Only] flash_encrypt_key_block_index = 0
 (Configures the index of the encryption key in the block_key. Default: 0.
 Range: 0~4. Note that this option can only be set to 0 for ESP32-C2. For more information, refer to the respective chip technical reference manual > Chapter eFuse Controller.)

• [SECURE OTHER CONFIG]

Other security configurations:

- flash_encryption_use_customer_key_enable = False (Configures whether to enable the use of a customer-specified encryption key)
- flash_encryption_use_customer_key_path = .\secure\flash_encrypt_key.bin(Path to the encryption key if it is used)
- flash_force_write_enable = False (Configures whether to skip encryption and secure boot checks during flashing. If it is set to False (default), an error message may pop up when attempting to flash products with enabled flash encryption or secure boot.)

• [FLASH ENCRYPTION KEYS LOCAL SAVE]

Encryption key related configurations:

- keys_save_enable = False (Configures whether to save the key)
- encrypt_keys_enable = False (Configures whether to encrypt the saved key)
- encrypt_keys_aeskey_path = (If you encrypt the key, please fill in the key file here, such as ./my_aeskey.bin)



• [ESP32* EFUSE BIT CONFIG]

Flash encryption items related configurations:

	Config Option	Description
	dl_encrypt_disable = False	Configures whether to disable encryption
[ESP32 DISABLE	dl_decrypt_disable = False	Configures whether to disable decryption
FUNC]	dl_cache_disable = False	Configures whether to disable cache
	jtag_disable = False	Configures whether to disable JTAG
	dis_usb_jtag = False	Configures whether to disable USB JTAG
	dis_pad_jtag = False	Configures whether to disable JTAG PAD
[ESP32-C* DISABLE	soft_dis_jtag = 7	Configures whether to soft-disable JTAG
FUNC]	dis_direct_boot = False	Configures whether to disable direct boot
	dis_download_icache = False	Configures whether to disable instruction cache in the Download mode
	dis_usb_jtag = False	Configures whether to disable USB JTAG
	hard_dis_jtag = False	Configures whether to hard-disable JTAG
	soft_dis_jtag = 7	Configures whether to soft-disable JTAG
[ESP32-S* DISABLE	dis_usb_otg_download_mode = False	Configures whether to disable USB OTG download
FUNC]	dis_direct_boot = False	Configures whether to disable direct boot
	dis_download_icache = False	Configures whether to disable instruction cache in the Download mode
	dis_download_dcache = False	Configures whether to disable data cache in the Download mode
	dis_direct_boot = False	Configures whether to disable direct boot
[ESP32-H* DISABLE	soft_dis_jtag = False	Configures whether to soft-disable JTAG
FUNC]	dis_pad_jtag = False	Configures whether to hard-disable JTAG
	dis_usb_jtag = False	Configures whether to disable USB JTAG

There will be a prompt message (shown below) when the tool is running. Check if the message is correct. The figure below shows the prompt message of enabling both flash encryption and secure boot:



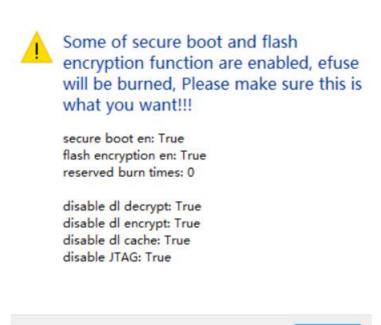


Figure 3-3. ESP32 Prompt Message of Enabling Flash Encryption and Secure Boot

During the firmware flashing process, the key and other information will be flashed into the chip's eFuse. After the flashing process is completed, "FINISH/完成" will be displayed.

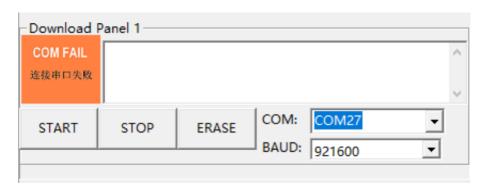
Prior to downloading, the tool verifies flash encryption and secure boot information in the eFuse , so as to prevent re-downloading to and damaging the encrypted module.



4. FAQ

4.1. COM Related Errors

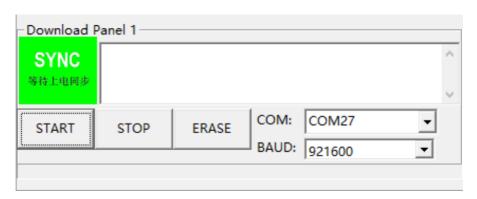
- 1. I cannot find the serial port that I am using in the *COM* drop-down menu when I open the Flash Download Tool.
 - A: First go to the device manager and check if the serial port has been successfully installed. If not, check the driver for any possible issues.
- 2. I get the "COM FAIL/连接串口失败" error message, as shown in the Figure below:



A: Firstly, make sure the selected COM is correct; Then, check if the COM is already occupied by another thread.

4.2. Synchronization Related Errors

1. The Flash Download Tool is stuck at the step shown in the figure below. How can I fix this?



- A: This may happen for the reasons given below.
- Hardware: The module is not in download mode.
- Software: The module selected in the tool is not the one you are actually using.

4.3. eFuse Related Errors

1. I click the **START** button, and get the error shown in the figure below.





A: You will get the "ESP8266 Chip efuse check error esp_check_mac_and_efuse" message when there are errors related to the eFuse. The possible causes are as follows:

- The eFuse is OK, but the module selected in the tool is not the one that is actually being used. In this situation, please select the module type based on your actual case.
- o There are problems with the eFuse of the module. In this case, please contact Espressif to obtain the required esptool.exe and operating instructions. Also, send the data that is read from eFuse to Espressif for further debugging.

4.4. Download Related Errors

- 1. Errors occur during downloading.
 - A: Please check the following:
 - o The TX/RX of the module is not used by other software programs.
 - o The module flash size is no less than the size of firmware to be downloaded.
 - If there is an MD5 verification error, erase the entire flash and try downloading again.

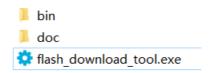
4.5. Operation Related Errors

- 1. The module crashes when powered on again after the firmware has been downloaded.
 - A: If the downloaded firmware works fine, then please check the following:
 - o The module selected in the tool is not the one you are actually using.
 - o The selected flash boot mode is wrong.
 - o The selected flash download mode is wrong.



Appendix A. Contents of the Flash Download Tool Folder

The figure below shows what the Flash Download Tool folder contains.



- doc folder: stores instruction documentation
- bin folder: stores firmware to be flashed
- flash_download_tool.exe: executable file of the Flash Download Tool.



Espressif IoT Team www.espressif.com

Disclaimer and Copyright Notice

Information in this document, including URL references, is subject to change without notice.

THIS DOCUMENT IS PROVIDED AS IS WITH NO WARRANTIES WHATSOEVER, INCLUDING WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. All liability, including liability for infringement of any proprietary rights, related to the use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

The Wi-Fi Alliance Member logo is a trademark of the Wi-Fi Alliance. The Bluetooth logo is a registered trademark of Bluetooth SIG.

All trade names, trademarks, and registered trademarks mentioned in this document are property of their respective owners and are hereby acknowledged.

Copyright © 2024 Espressif Inc. All rights reserved. All rights are reserved.