ESP32-P4

User Guide of ESP Test Tools

ESPRESSIF

Table of contents

This documentation for ESP32-P4 currently focuses on Flash Download Tool usage. It includes the user guide, frequently asked questions, related resources, and legal notices.

1 Flash Download Tool User Guide

1.1 Preparation

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

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

1.2 Tool Overview

User Interface

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

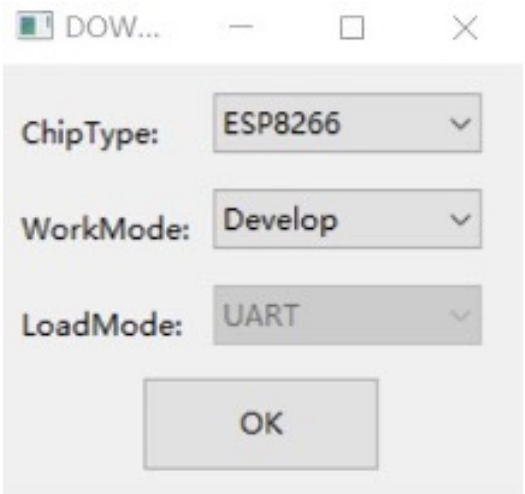

Fig. 1: Flash Download Tool Main Interface

- `ChipType`: Selects the chip type for your product.
- `WorkMode`: Work mode of the tool. Below are the differences between the two modes supported currently, `Develop` and `Factory` modes.
  - `Develop` 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 be flashed in the bin folder at the same level as the `.exe` file. 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: Supports both UART and USB

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
  - SPI SPEED: SPI boot rate
  - SPI MODE: SPI boot mode
  - 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.
  - Default button: restores the SPI configuration to the default values.
- Download Panel
  - START: Starts downloading
  - STOP: Stops downloading
  - ERASE: Erases the entire flash
  - COM: Serial port used for downloading
  - BAUD: Baud rate

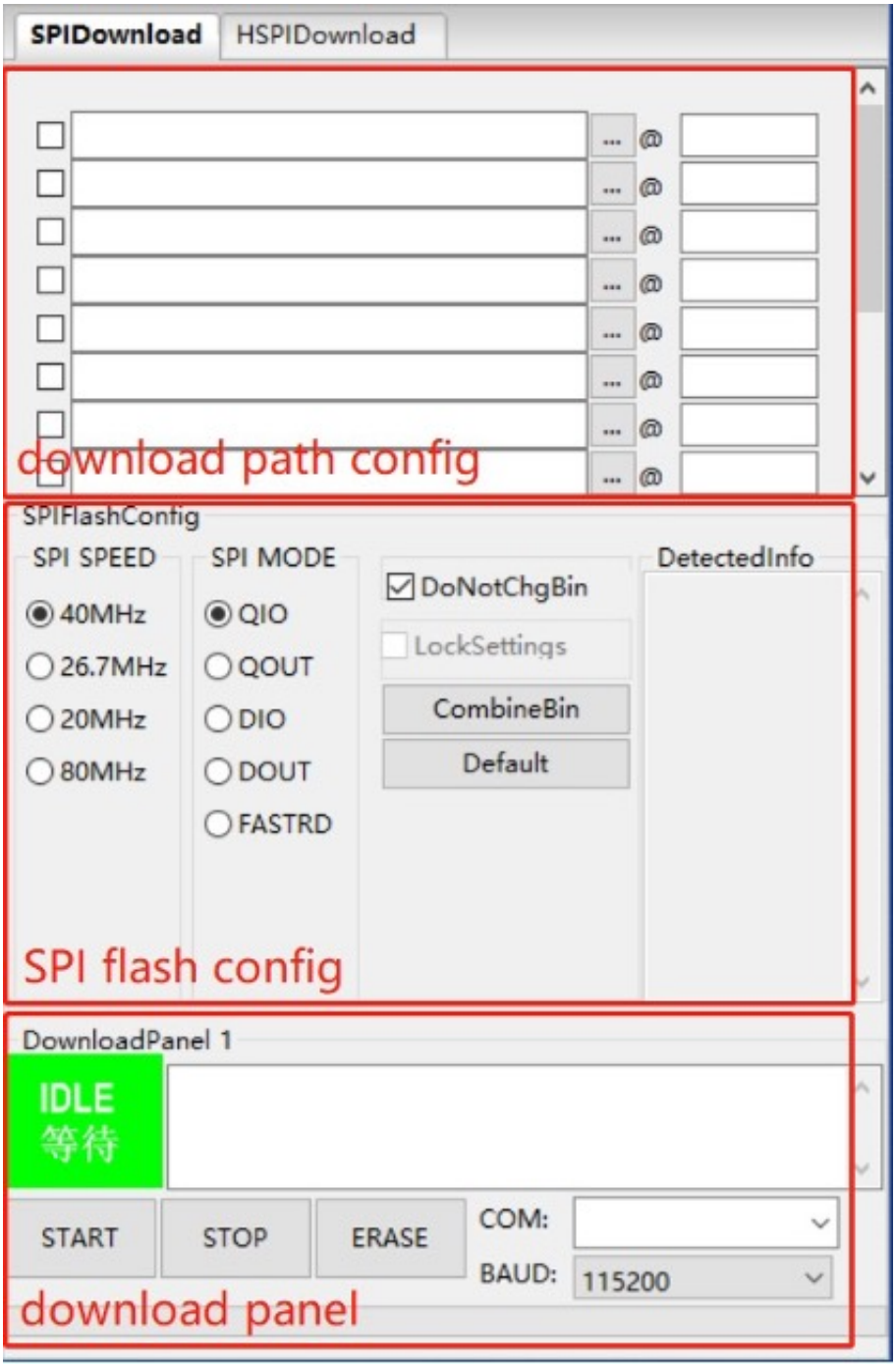

Fig. 2: SPIDownload Tab

Note:

- When the tool version is >= 3.9.10, `SPI SPEED` and `SPI MODE` cannot be changed by default. These parameters are consistent with the firmware build configuration.

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.)
- In `Factory` mode, you can click the `CRC32 cal` button to generate a checksum based on all files to be flashed and their flash addresses. The production line can use this checksum to verify that the files and configuration to be flashed are correct.

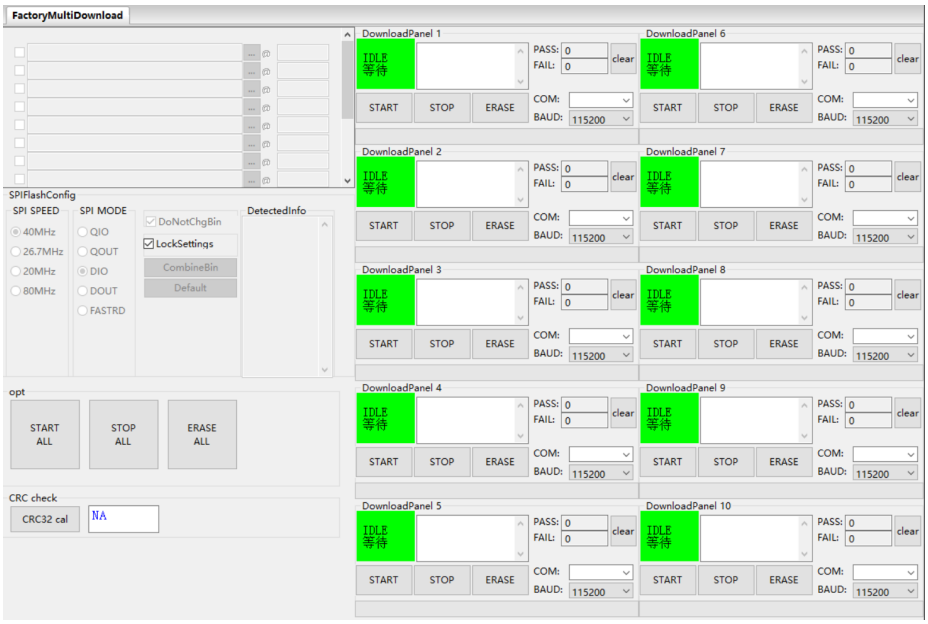

Fig. 3: FactoryMultiDownload Tab

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

chipInfoDump Tab

- `Device`: Selects the device's serial port number and communication baud rate.
- `Read Flash`: Specifies the start address and size of the content to be read from the flash. This setting is only required when reading flash.

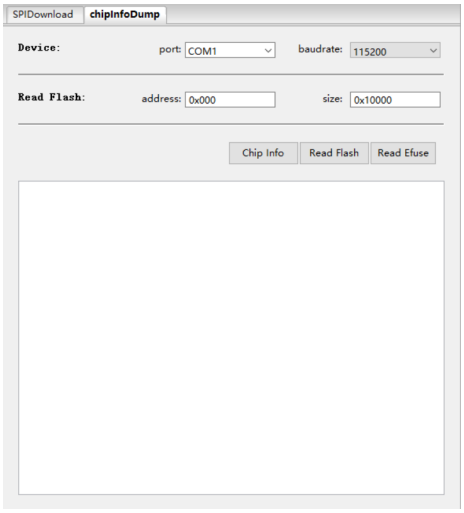

Fig. 4: chipInfoDump Tab

- Function Description
  - `Chip Info`: Reads the chip model, flash ID, and flash status register values. The read content is displayed directly in the tab.
  - `Read Flash`: Reads data stored in the flash. The read content is saved in a generated bin file, which is named in the format "Chip MAC + Start Address of Reading + Data Length of Reading + Reading Time".
  - `Read Efuse`: Reads the chip's eFuse content, with functionality identical to *esptool summary*. The read content is stored in a generated text file, named in the format "Chip MAC + Reading Time".

Note:

- To use the above reading functions, the product should enter download mode after startup.
- Tool version >= 3.9.8

1.3 Download Example

This section takes the ESP32 series as an example to demonstrate how to perform both regular and encrypted download operations. At present, ESP32-P4 only supports regular download, and will support encrypted download later.

Regular Download

1. Pull GPIO35 low and GPIO36 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`

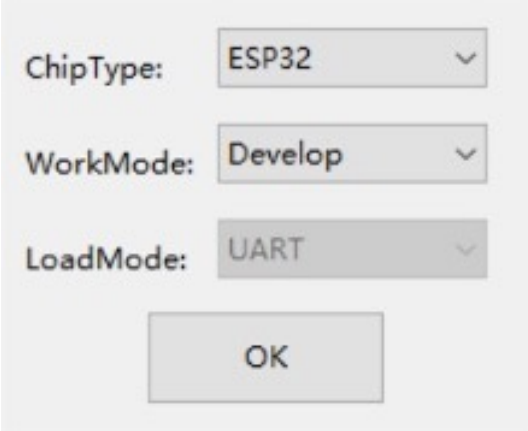

Fig. 5: Selecting Device —ESP32 Download Tool

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 enter `COM` and `BAUD` according to your requirements.
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 the following figure.

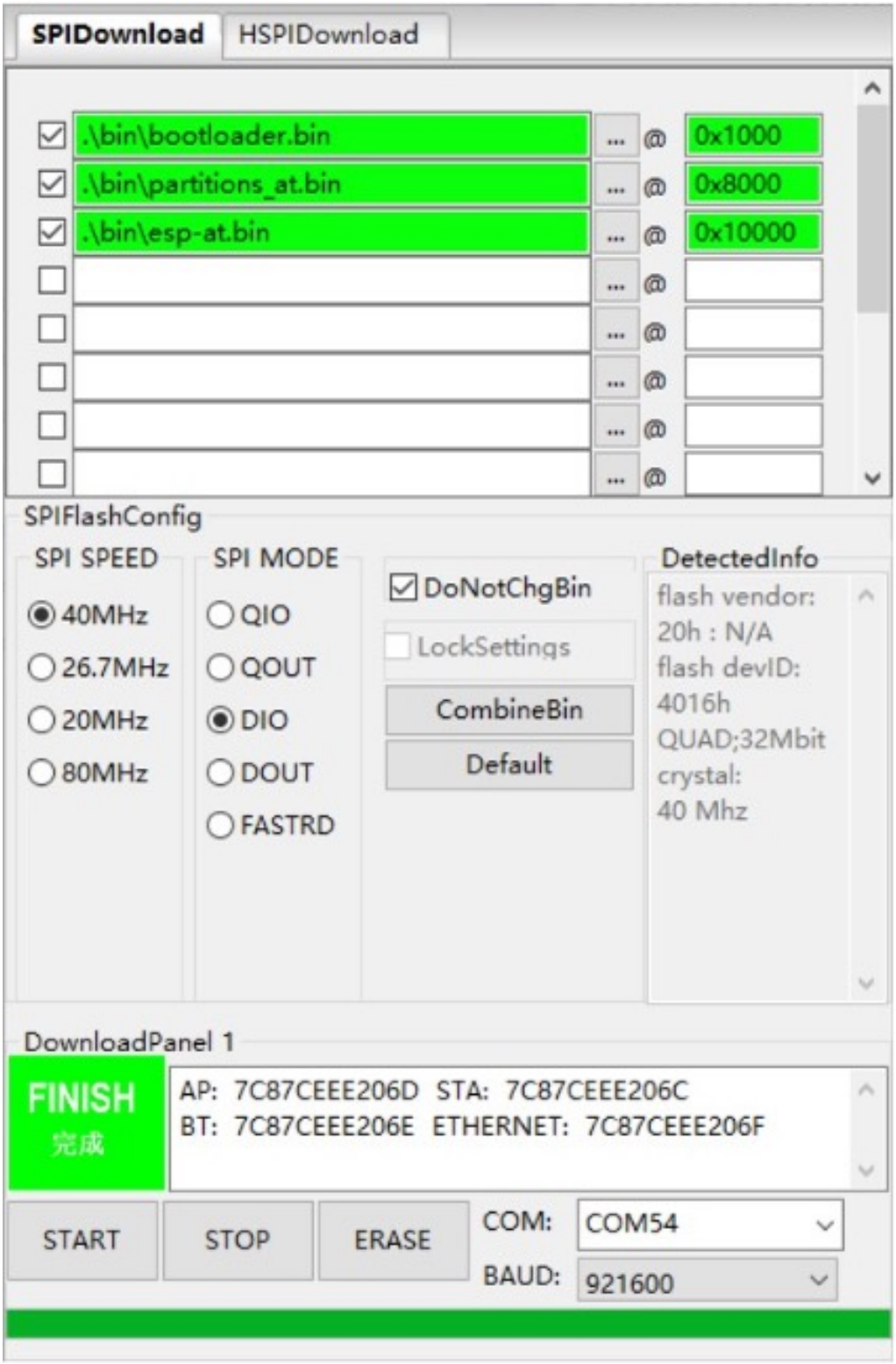

Fig. 6: Download Completed

Encrypted Download

The encrypted firmware downloading process is as follows:

1. Flash Download Tool downloads the plaintext firmware to the chip.
2. The chip uses the key in its eFuse to encrypt the firmware and write it to the flash.

3. If there is no such key in the eFuse, the tool will automatically generate a random one and flash it to eFuse. You can also prepare your own encryption key. If there is, the tool skips the key generation and flashing process.

To configure the encryption function, follow the steps below:

1. Open the configuration file ./configure/[chip_name]/security.conf. If there is no such file, for example, when you open the tool for the first time, restart the tool.
2. 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)
  - **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 = 1** (Configures the number of reserved flashing operations. Optional values: 0, 1.)
  - **flash_encrypt_key_block_index = 0** (Configures the index of the encryption key in the block_key. Default: 0. Range: 0~4. For more information, refer to Technical Reference Manual (PDF) > Chapter eFuse Controller.)
- **[SECURE OTHER CONFIG]** Other security configurations:
  - **flash_encryption_use_customer_key_enable = False** (Configures whether to enable a customer-specified encryption key)
  - **flash_encryption_use_customer_key_path = .\secure\flash_encrypt_key.bin** (If using a customer-specified key, the key path needs to be specified here.)
  - **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]** Determines whether to store the encryption key file locally. Default: False.
- **keys_save_enable = False** (Configures whether to save the key.)
- **encrypt_keys_enable = False** (Configure whether to encrypt the locally stored key.)
- **encrypt_keys_aeskey_path =** (If you encrypt the locally stored key, please fill in the key file here, such as **./my_aeskey.bin**)
- **[ESP32* EFUSE BIT CONFIG]** Determines whether to set encryption items when flash encryption is enabled. Default: False.

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:

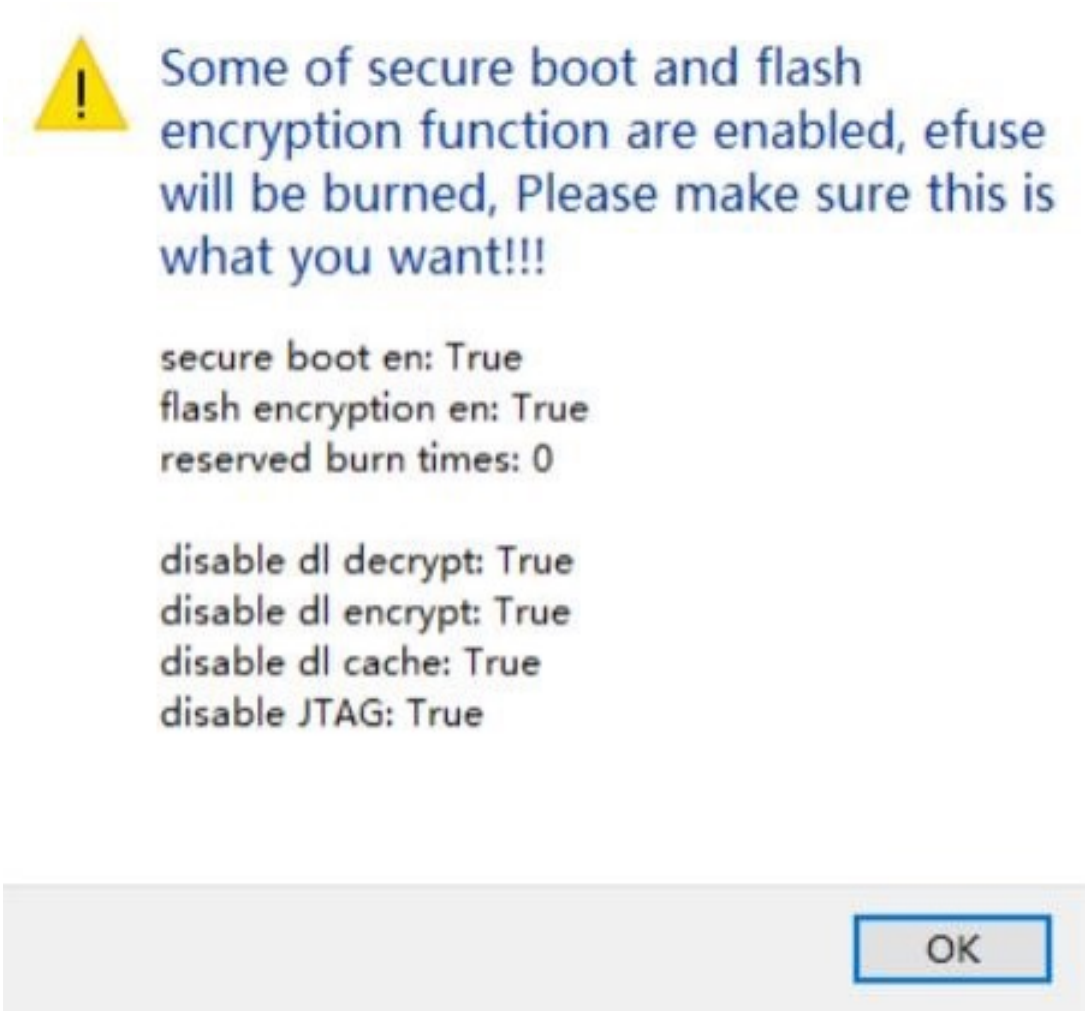

Fig. 7: ESP32 Prompt Message of Enabling Flash Encryption and Secure Boot

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.

Note: 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.

1.4 Version Release Notes

- 3.9.10
  - Support ESP32-P4 ECO6
  - Support ESP32-C5 ECO3
  - `SPI_MODE` and `SPI_SPEED` are not editable
  - Support CRC32 check for files to be flashed and flash addresses in factory mode
  - Support ESP32-C5 secure boot V2 and flash encryption
  - Support ESP32, ESP32-S2, and ESP32-S3 eFuse flash_voltage configuration
- 3.9.9
  - Support ESP32-P4 USB download
  - Add a help button for tool updates and online documentation
  - Supports flashing ESP32-P4 devices with flash larger than 16 MB
  - Add MD5 verification status display during flashing

- 3.9.8
  - Add `Read Flash` and `Read Efuse` tabs
  - Support ESP32-C5 and ESP32-C61
- 3.9.7
  - Support ESP32-P4
  - Update UI display icons
  - Fix icon stuttering during flashing after long-term use
- 3.9.6
  - Support secure boot V2 and flash encryption for ESP32, ESP32-H2, ESP32-C6, ESP32-C2, and ESP32-S2
  - In factory mode, the number of products to be flashed is configurable, up to 20
- 3.9.5
  - Support ESP32-H2
  - Support disabling the erase button
  - Fix some known vulnerabilities
- 3.9.4
  - Support ESP32-C6
  - Support ESP32-C3 and ESP32-S3 flash encryption
  - Add XMC flash fix
  - Support flashing count in factory mode
- 3.9.3
  - Support ESP32-C2
  - Update the icon shown when flashing completes
- 3.9.2
  - Fix configuration file errors
  - Update the auto-start process

2 FAQ

- *Flash Download Tool FAQs* cover common questions about the *Flash Download Tool User Guide*.

2.1 Flash Download Tool

1. I cannot find the serial port in the COM drop-down menu of the Flash Download Tool.

First, check the Device Manager to ensure the serial port is properly installed. If not, check the driver for any issues.

2. I got a "COM FAIL" error, as shown below:

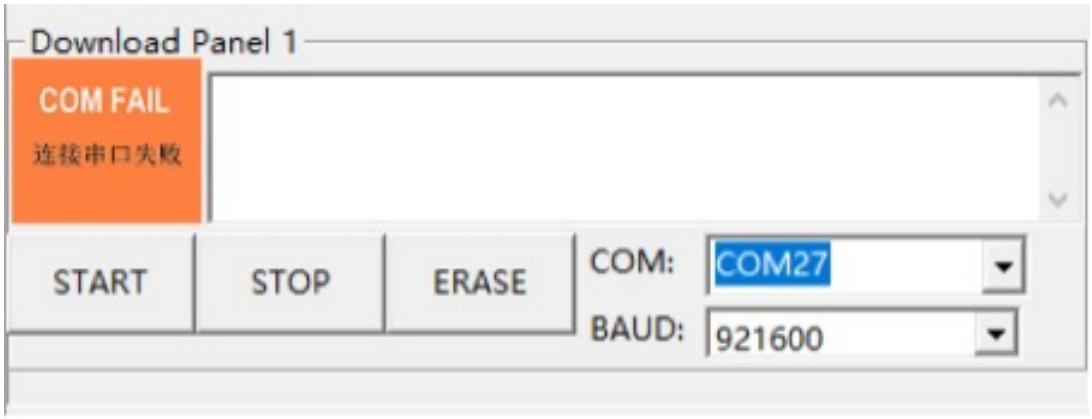

Fig. 8: Connection Failure of Serial Port

- Make sure the correct COM port is selected
- Verify that the COM port isn't being used by another thread.

3. The Flash Download Tool is stuck, as shown in the figure below. How can I fix this?

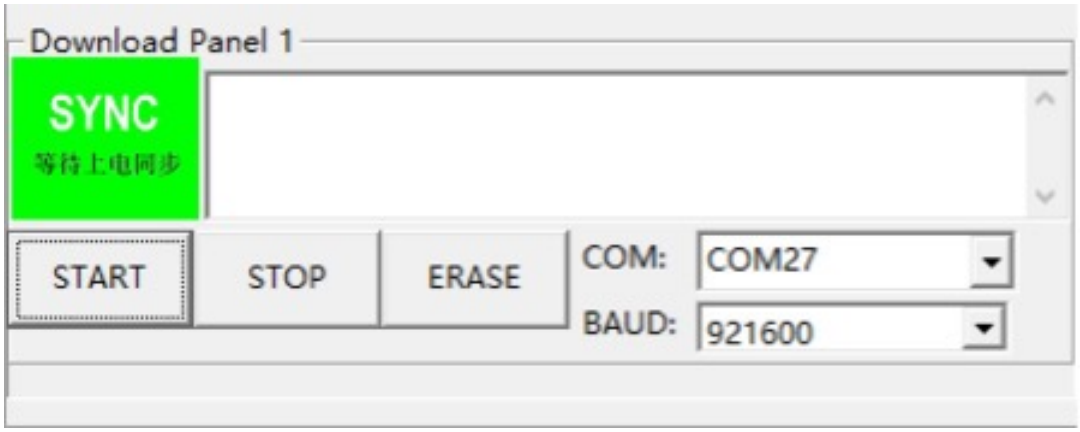

Fig. 9: Download Panel

This issue may occur due to:

- Hardware: The module is not in download mode.
- Software: The module selected in the tool isn't the one you are using.

4. I clicked the START button and got the error shown below.

Fig. 10: eFuse Error

The `ESP8266 Chip efuse check error esp_check_mac_and_efuse` message indicates an issue with the eFuse. Possible causes include:

- The eFuse is fine, but the wrong module is selected in the tool. In this case, select the correct module based on your setup.

- There is a problem with the module' s eFuse. In this case, contact Espressif for esptool.exe and instructions, and send the eFuse data to Espressif for further debugging.

5. Errors occur during downloading.

Check the following:

- Ensure the module' s TX/RX pins are not used by other software.
- Make sure the module' s flash size is no less than the firmware size.
- If you encounter an MD5 verification error, erase the entire flash and try downloading again.

6. The module crashes when powered on again after downloading the firmware.

If the firmware works correctly, check the following:

- The module selected in the tool matches the one you are using.
- The correct flash boot mode is selected.
- The correct flash download mode is selected.

3 Related Documentation and Resources

- Chip Datasheet (PDF)
- Technical Reference Manual (PDF)
- Chip Variants
- Modules
- Development Boards
- ESP Product Selector
- Regulatory Certificates
- User Forum (Hardware)
- Technical Support

4 Disclaimer and Copyright Notice

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

All third party' s information in this document is provided as is with no warranties to its authenticity and accuracy.

No warranty is provided to this document for its merchantability, non-infringement, fitness for any particular purpose, nor does any warranty otherwise arising out of any proposal, specification or sample.

All liability, including liability for infringement of any proprietary rights, relating to 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.