

ESP32-C61

User Guide of ESP Test Tools



ESPRESSIF

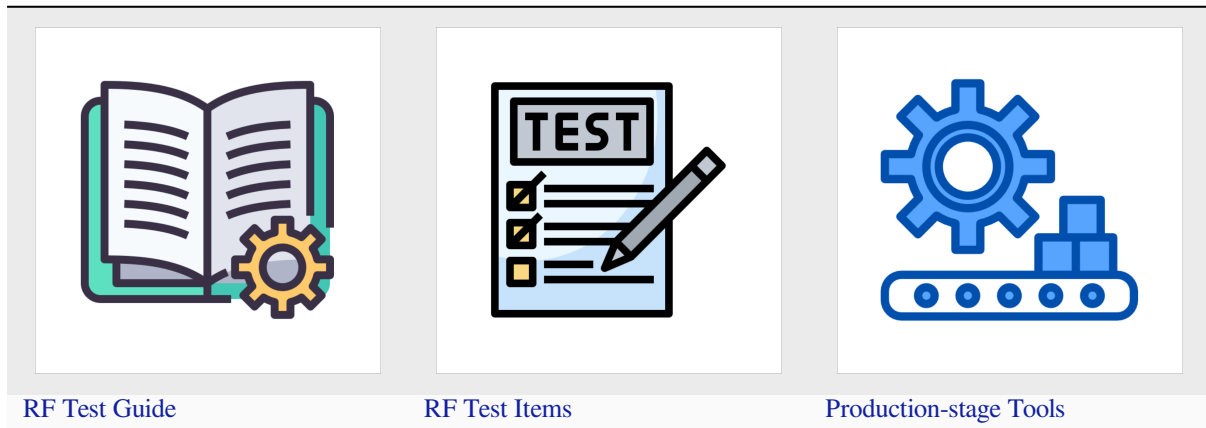
Table of contents

Table of contents	i
1 Development Stage	1
1.1 RF Test Tool	1
1.2 RF Test Items	1
1.3 RF Certifications	2
1.4 WFA Certification and Testing Guideline	2
2 EspRFTesTool Toolkit	2
2.1 EspRFTesTool	3
2.2 DownloadTool	6
2.3 PowerLimitTool	8
3 RF Test Items	16
3.1 Wi-Fi Non-Signaling Test	16
3.2 Wi-Fi Signaling Test	22
3.3 Wi-Fi Adaptivity Test	25
3.4 Wi-Fi Blocking Test	31
3.5 Bluetooth LE Non-Signaling Test	35
3.6 Bluetooth LE DTM Test	41
3.7 Bluetooth LE Adaptivity Test	45
3.8 Bluetooth LE Blocking Test	48
4 RF Test Certification	48
4.1 CE Certification	48
4.2 FCC Certification	49
4.3 SRRC Certification	49
5 WFA Certification and Testing Guide	49
5.1 Overview	49
5.2 Introduction to WFA Certification	49
5.3 Espressif Product Certification Process	50
5.4 WFA Testing	53
6 Production Stage	61
7 Flash Download Tool User Guide	61
7.1 Preparation	61
7.2 Tool Overview	62
7.3 Download Example	65
8 Espressif Production Testing Guide	69
8.1 Introduction	70
8.2 Environment Setup	75
8.3 Production Testing Tool	76
8.4 Appendix B: GPIO Conductivity Test Configuration	84
8.5 Appendix C: Firmware Version Verification Test	85
8.6 Certification	85
9 Test Fixture Manufacturing Instruction	85
9.1 About This Instruction	85
9.2 Overview	86
9.3 The Main Structure of a Typical Module Fixture	86
9.4 Fixture Testing	97
9.5 Appendix	101
10 Matter QR Code Generator	102
10.1 Software Directory	102
10.2 Get Started	102
10.3 Start Printing	104
10.4 Check Printed Labels	106
10.5 Integrate Laser Marking	107
10.6 Appendix I: Flash Scan Board Firmware	107
10.7 Appendix II: BarTender (2022) Installation Process	108
11 FAQ	111
11.1 RF Testing	111

11.2	WFA Certification Test	111
11.3	Flash Download Tool	112
11.4	Espressif Production Testing Guide	113
12	Related Documentation and Resources	114
13	Disclaimer and Copyright Notice	114

This repository provides comprehensive resources to support the development and production of products based on [Espressif chips](#) and [modules](#).

For the development stage, it provides an RF testing tool and detailed test guidelines to ensure your product meets the necessary performance and certification standards. Additionally, for the production stage, the repository includes essential tools and instructions to streamline the manufacturing process, ensuring efficient testing, validation, and quality control of your products.



1 Development Stage

To ensure your product meets requirements for related [RF Certifications](#), this repository provides the testing tools and guidelines to facilitate RF testing, ensuring compliance with global standards and industry certifications.

1.1 RF Test Tool

[EspRFTestTool Toolkit](#) is a comprehensive tool that allows you to control devices and evaluate key RF performance metrics. It supports the following [RF Test Items](#).

1.2 RF Test Items

Wi-Fi Test

- [Wi-Fi Non-Signaling Test](#) also known as fixed frequency test, directly controls the device to transmit specific signals without establishing a data connection. It evaluates key RF performance metrics, such as transmit power, spectrum quality, and error rate, ensuring wireless communication quality in various scenarios.
- [Wi-Fi Signaling Test](#) assesses and verifies the Wi-Fi signaling functions of wireless network devices, focusing on stable and reliable communication across different operating scenarios. It evaluates the Over-The-Air (OTA) performance, including Total Radiated Power (TRP) and Total Isotropic Sensitivity (TIS).
- [Wi-Fi Adaptivity Test](#) simulates various network conditions and loads to access device's real-time adjustments in transmission rate, channel selection, and power levels, optimizing overall network performance and stability.
- [Wi-Fi Blocking Test](#) evaluates the device's reception performance in environments with strong interference. By introducing high-intensity interference signals, it measures reception sensitivity and anti-interference capability, ensuring reliable operation in complex wireless environments.

Bluetooth Test

- *Bluetooth LE Non-Signaling Test* controls the device to transmit specific signals without establishing a connection, evaluating performance metrics such as transmit power, spectrum characteristics, and error rate to ensure communication quality.
- *Bluetooth LE DTM Test* evaluates the RF performance of Bluetooth LE devices by directly controlling the device to enter specific transmission or reception modes, accessing key metrics like transmit power, reception sensitivity, and spectrum characteristics.
- *Bluetooth LE Blocking Test* assesses device stability and performance in environments with interference from other wireless signals, ensuring compliance with relevant standards.
- *Bluetooth LE Adaptivity Test* ensures the device meets performance criteria during frequency hopping, particularly when the Power Spectral Density (PSD) of the Bluetooth LE signal exceeds 10 dBm/MHz, avoiding interference with other wireless devices.

1.3 RF Certifications

The *RF Test Items* outlined above are designed to ensure your product complies with the standards required for the following certifications:

- *CE Certification*: A mandatory certification by the EU, confirming compliance with safety, health, and environmental protection standards.
- *FCC Certification*: A mandatory certification by the U.S. Federal Communications Commission, ensuring compliance with regulations on radio spectrum use, electromagnetic compatibility, and RF radiation.
- *SRRC Certification*: A mandatory certification for radio equipment in China, ensuring compliance with national radio management regulations to avoid electromagnetic interference.

Test items for each certification are listed in the following table.

Table 1: Test Items for RF Certifications

	CE Certification	FCC Certification	SRRC Certification
Wi-Fi Non-Signaling Test	Y	Y	Y
Wi-Fi Adaptivity Test	Y	—	Y
Wi-Fi Blocking Test	Y	—	—
Bluetooth LE Non-Signaling Test	Y	Y	Y
Bluetooth LE DTM Test	Y	—	—
Bluetooth LE Adaptivity Test	Y	—	—
Bluetooth LE Blocking Test	Y	—	—

Note: *Wi-Fi Signaling Test* is not typically required for standard RF certifications; it is primarily used to evaluate the OTA performance of devices.

1.4 WFA Certification and Testing Guideline

In addition, this repository also provides *WFA Certification and Testing Guide*, which provides detailed information about the WFA certification process and testing requirements to help you pass the Wi-Fi Alliance certification.

2 EspRFTTestTool Toolkit

The **EspRFTTestTool toolkit** is an RF test tool provided by Espressif. It contains EspRFTTestTool, DownloadTool, and PowerLimitTool.

- *EspRFTTestTool*: Used to perform RF tests;
- *DownloadTool*: Used to download the firmware required for RF tests;
- *PowerLimitTool*: Used to generate customized phy_init_data firmware.

Download Link: [EspRFTTestTool toolkit](#)

The zip file not only includes the EspRFTTestTool toolkit but also contains all the necessary firmware for *RF Test Items*, allowing users familiar with the testing process to directly use the firmware for testing.

Note: In this document, the **EspRFTTestTool toolkit** refers to the collection of the three tools, while the **EspRFTTestTool** refers to this single tool.

2.1 EspRFTTestTool

The main interface of the EspRFTTestTool toolkit is the EspRFTTestTool, which includes the COM Port Configuration area, the Download Configuration area, the RF Test Configuration area, and the Log window.

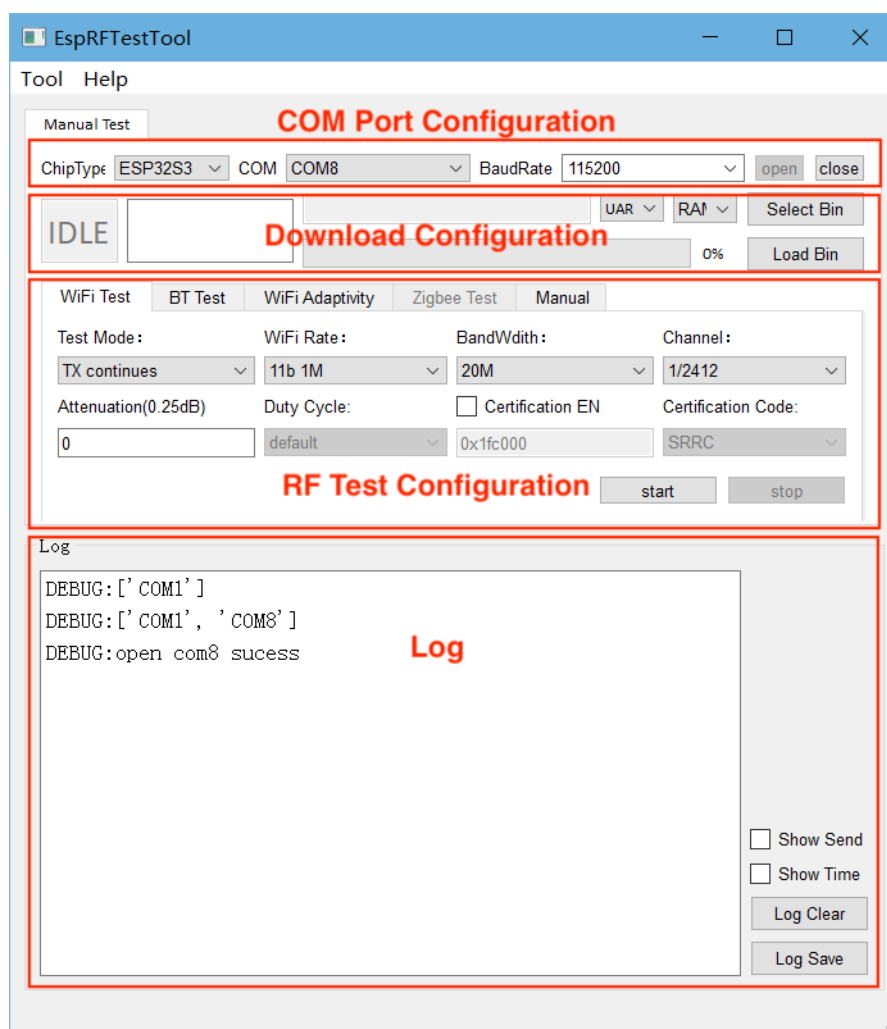


Fig. 1: EspRFTTestTool

COM Port Configuration Area

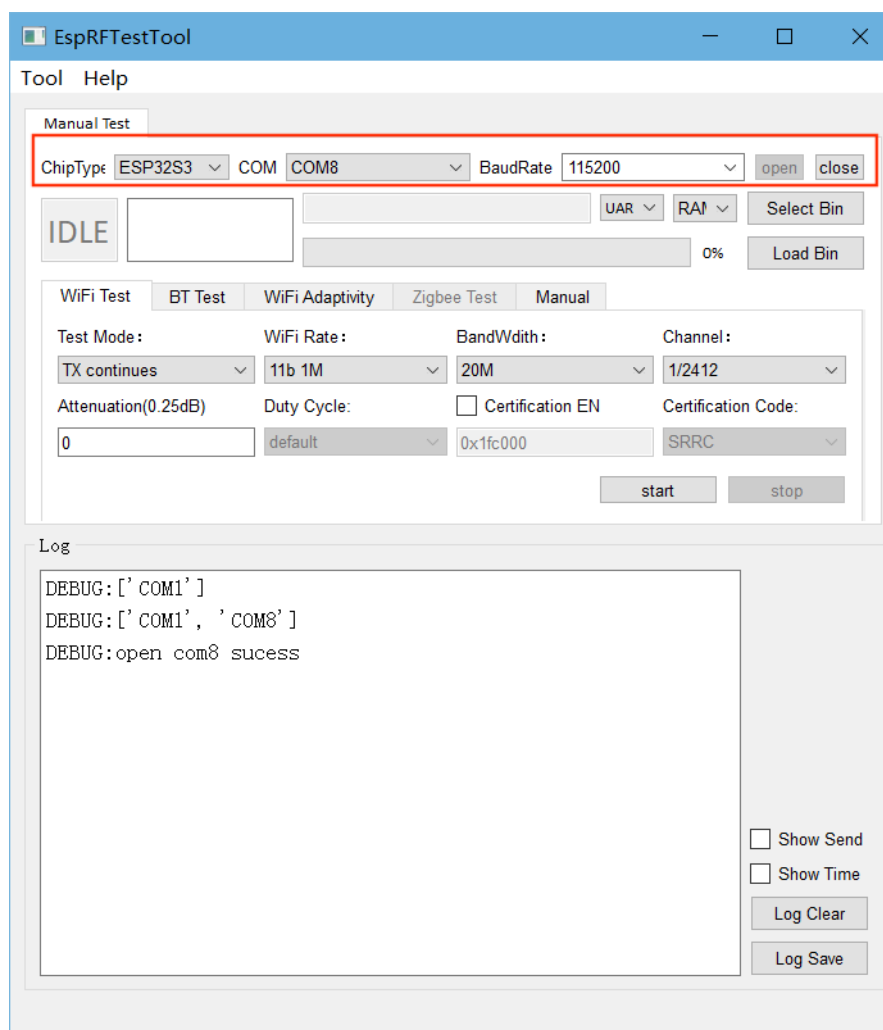


Fig. 2: EspRFTTestTool COM Port Configuration Area

- **ChipType:** Select the chip;
- **COM:** Select the serial port number;
- **BaudRate:** Select the baud rate;
- **Open:** Open the serial port;
- **Close:** Close the serial port.

After configuring the serial port, you can perform quick flashing and RF tests.

Download Configuration Area

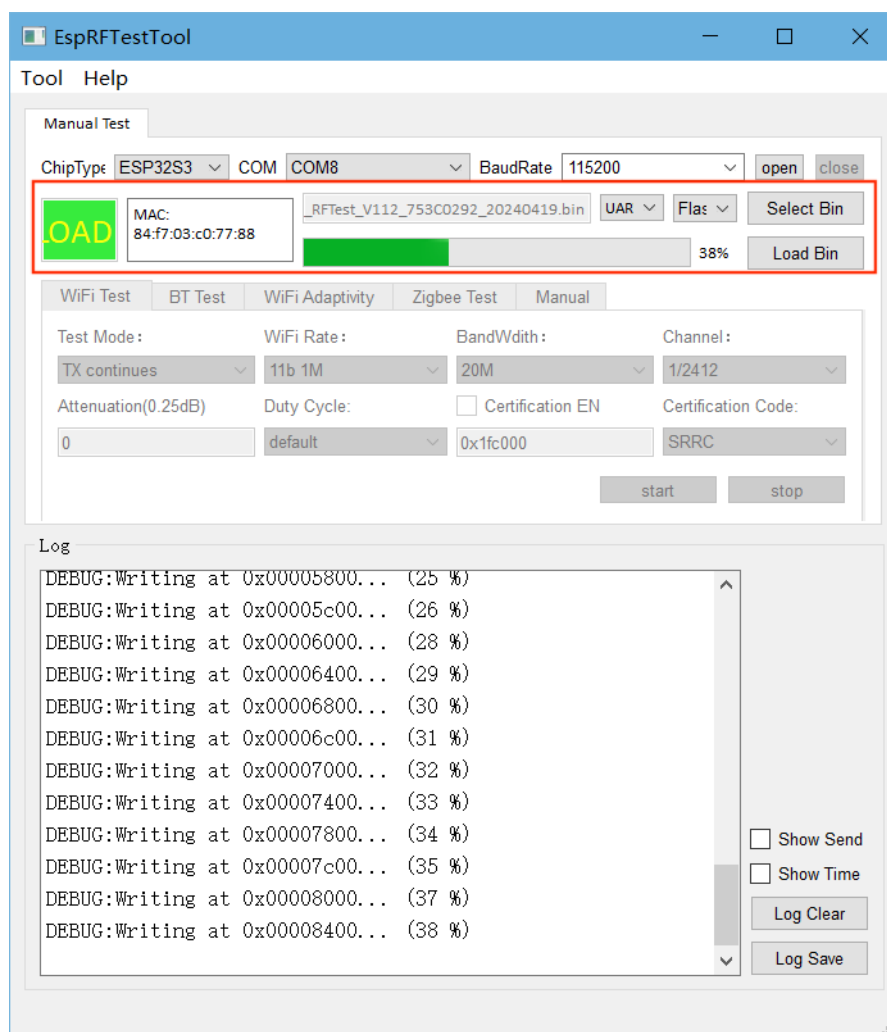


Fig. 3: EspRFTTestTool Download Configuration Area

Generally, the [DownloadTool](#) is used to download the firmware required for RF tests. However, for some simple firmware, such as non-signaling test firmware and adaptivity test firmware, EspRFTTestTool can be used for quick flashing.

- Pull down the Boot pin and re-power the chip to enter download mode;
- By default, flashing is conducted through UART;
- Select flash to download to the flash;
- Click **Select Bin** to select the bin file to be flashed;
- Click **Load Bin** to start flashing;
- After flashing is completed, pull up the Boot pin and re-power the chip to enter operation mode.

RF Test Configuration Area

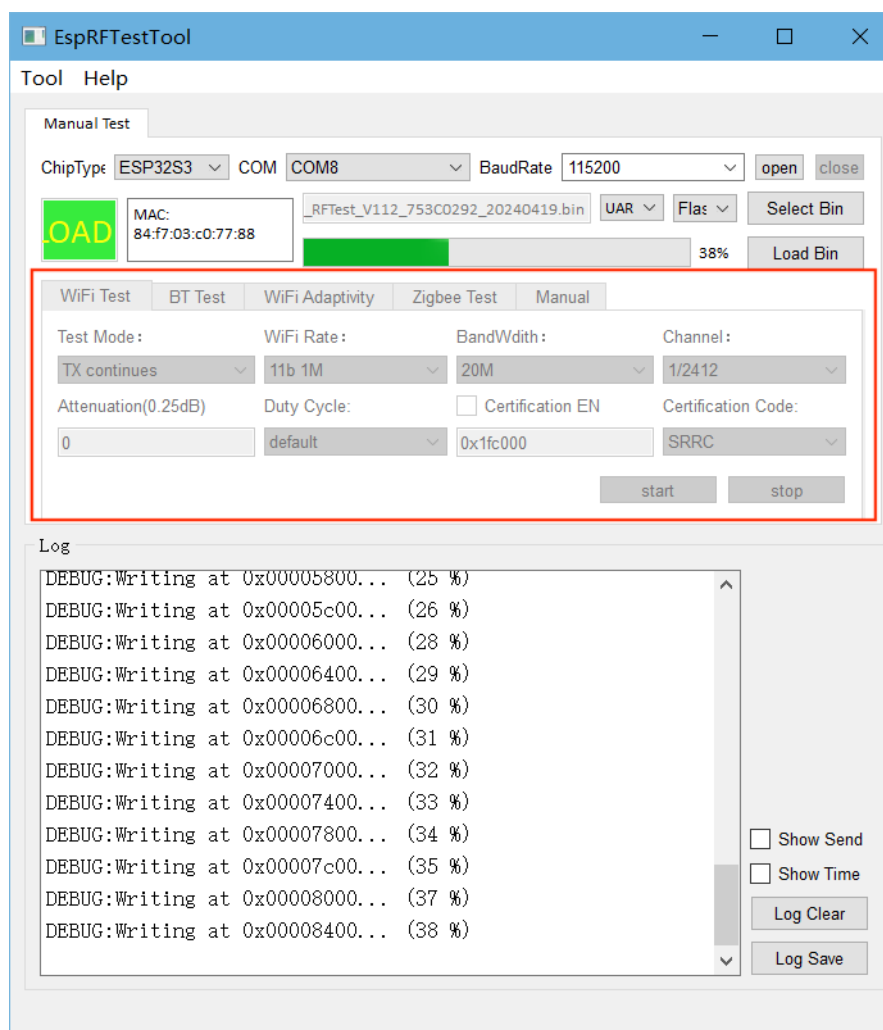


Fig. 4: EspRFTesTool RF Test Configuration Area

After flashing the firmware, you can perform the corresponding RF tests:

- **Wi-Fi Test:** Used for Wi-Fi Non-Signaling Test;
- **BT Test:** Used for Bluetooth and Bluetooth LE Non-Signaling Test;
- **Wi-Fi Adaptivity:** Used for Wi-Fi Adaptivity Test;
- **Zigbee Test:** Used for 802.15.4 Non-Signaling Test;
- **Manual:** Used to enter serial port commands.

For specific parameter configuration, please refer to the corresponding RF test document.

Log Window

The Log window is used to display the status of the tool. To view the log printed via the chip serial port, please use a general serial port assistant, such as [SerialPortUtility](#).

2.2 DownloadTool

Click **Tool** in the toolbar and select **DownloadTool** to enter the DownloadTool interface.

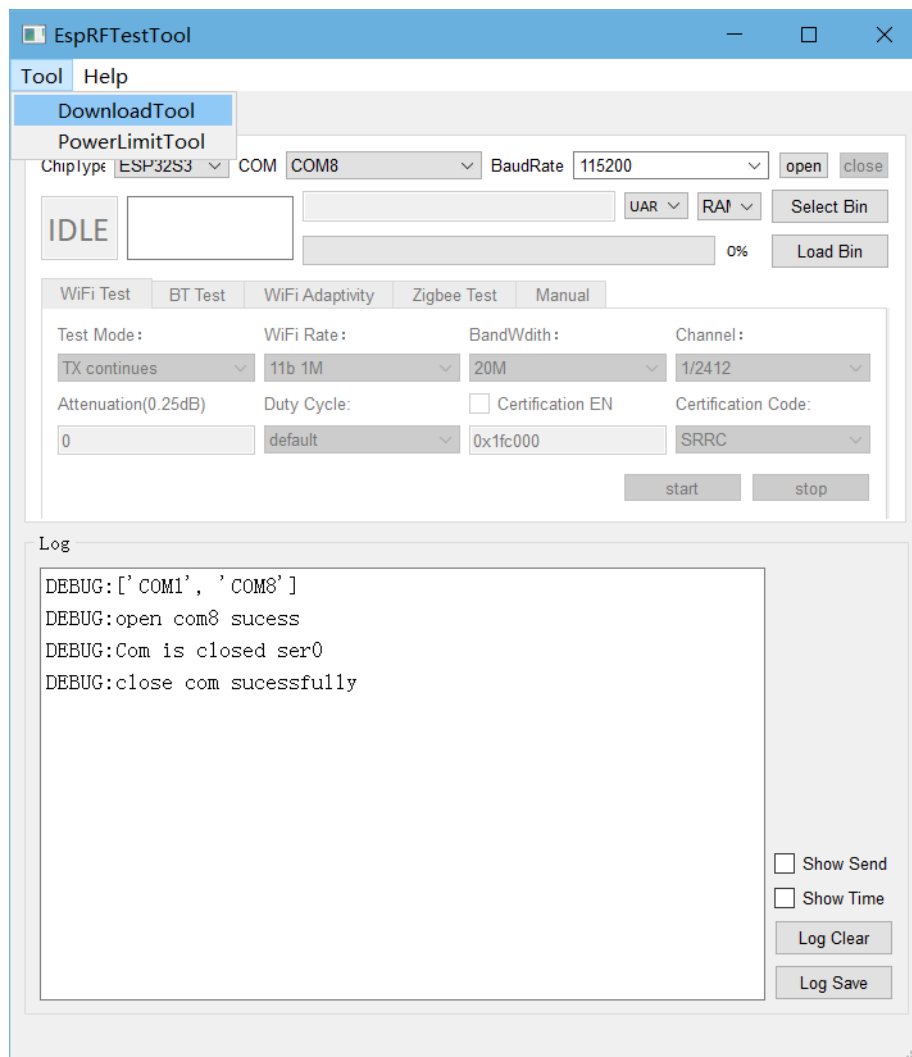


Fig. 5: Entry to DownloadTool

Follow the steps below to flash the firmware:

- Set the Chip Type, COM Port, and Baud Rate. Then, click Open to open the serial port;
- Select flash to download to the flash;
- Select the firmware and flash it to the specified address;
- Check whether the chip has entered download mode. If yes, click Start Load to start flashing. After flashing is completed, the SUCC sign shows up;
- After flashing is completed, click Close to close the serial port.

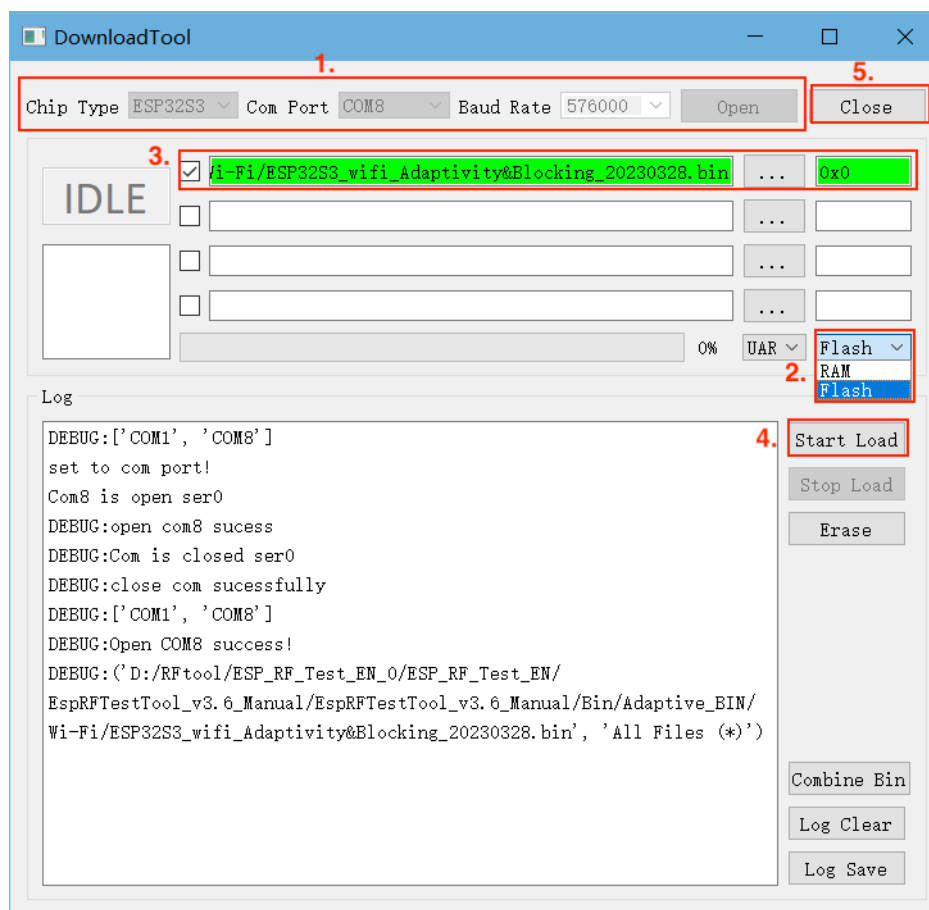


Fig. 6: DownloadTool Interface

Note: How to check whether the chip has entered download mode:

1. Close the serial port of DownloadTool and open a general serial port assistant, such as [SerialPortUtility](#);
2. Configure the serial port number and baud rate, pull down the Boot pin, re-power the chip, and the serial port assistant will print the log like waiting for download;
3. Close the serial port assistant, open DownloadTool, and start flashing;
4. After the flashing is completed, pull up the Boot pin, and re-power the chip to enter operation mode. If there are any abnormal behaviors, use the serial port assistant to check.

Note: By default, DownloadTool flashes to RAM. To specify a flash address, you need to switch to flashing to flash first.

2.3 PowerLimitTool

PowerLimitTool generates single-country and multi-country phy_init_bin files by configuring Wi-Fi output power to ensure your products meet the regulatory requirements of different countries or regions.

Note: The following methods can be used to limit Wi-Fi power. If multiple methods are used together, the minimum power value will be taken:

1. Use the API (`esp_wifi_set_max_tx_power`) to limit the maximum output power.

2. Configure Max Wi-Fi TX Power in Menuconfig, which serves the same function as the API mentioned above and can limit the maximum output power.
3. Use the Phy Init Bin function to modify the phy_init_data.h file in ESP-IDF.
4. Use the Phy Init Bin function to generate the phy_init_data.bin file by referring to the introduction in this document.

Under the main interface of EspRFTTestTool, click Tool, and select PowerLimitTool from the dropdown box to open PowerLimitTool.

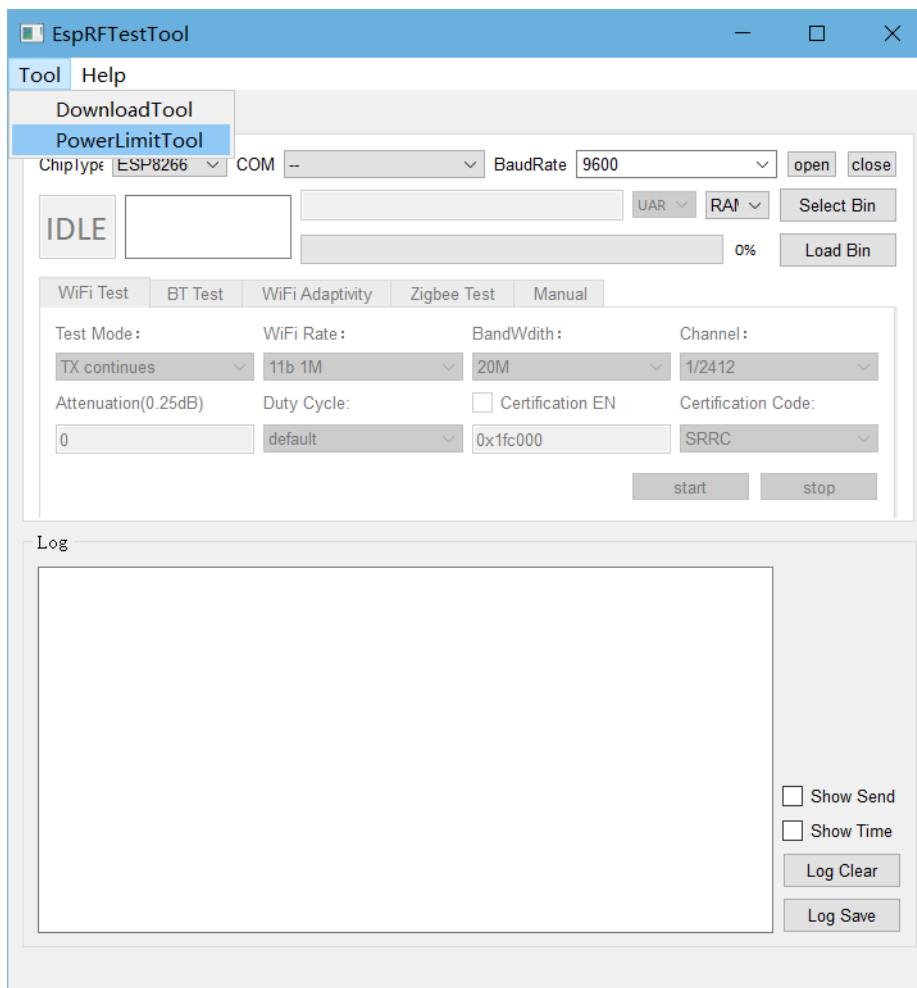


Fig. 7: Entry to PowerLimitTool

1. In the main interface of PowerLimitTool, click the Chip dropdown box to view the chips supported by the tool and select a chip (This section takes ESP32-C3 as an example).

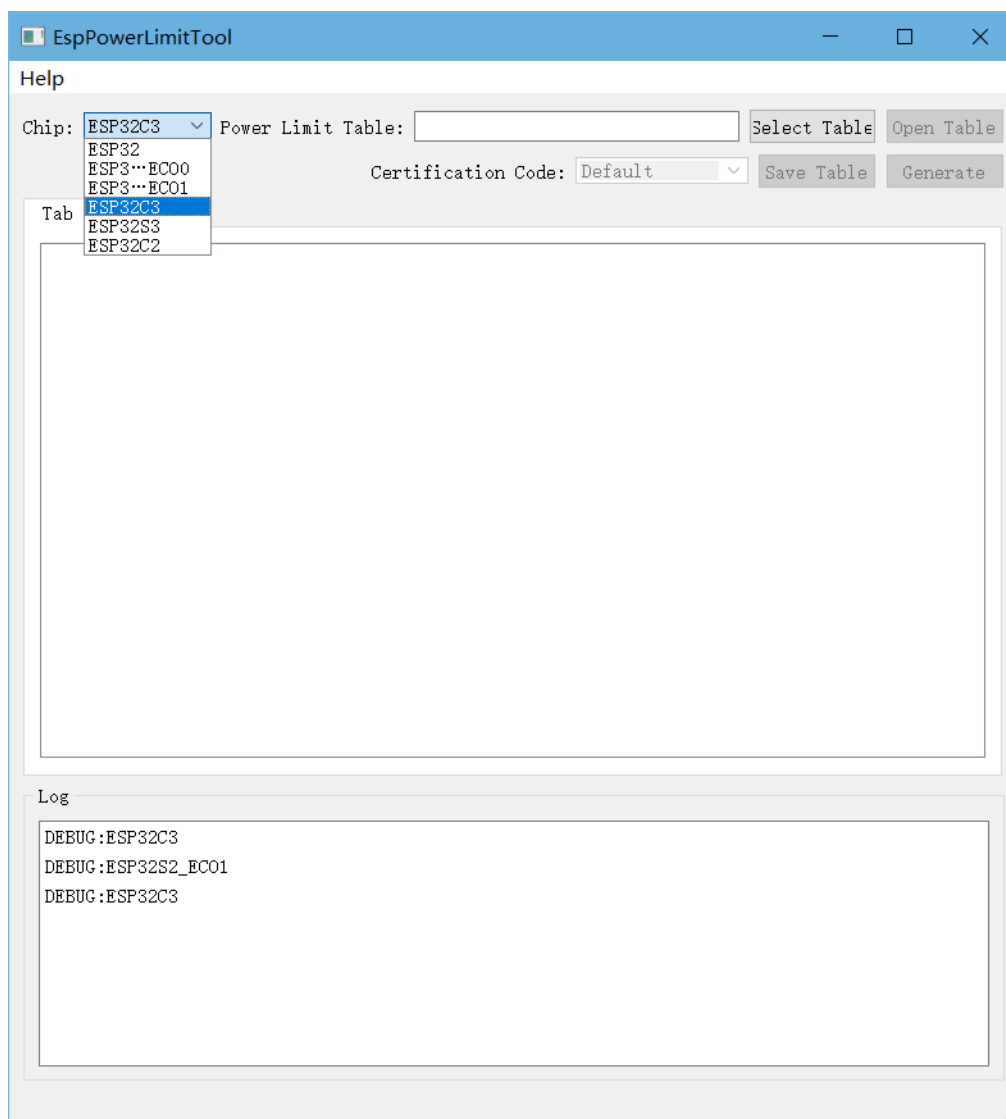


Fig. 8: PowerLimitTool Main Interface

2. Click `Select Table` and select the TX Power Setting table for your chip.

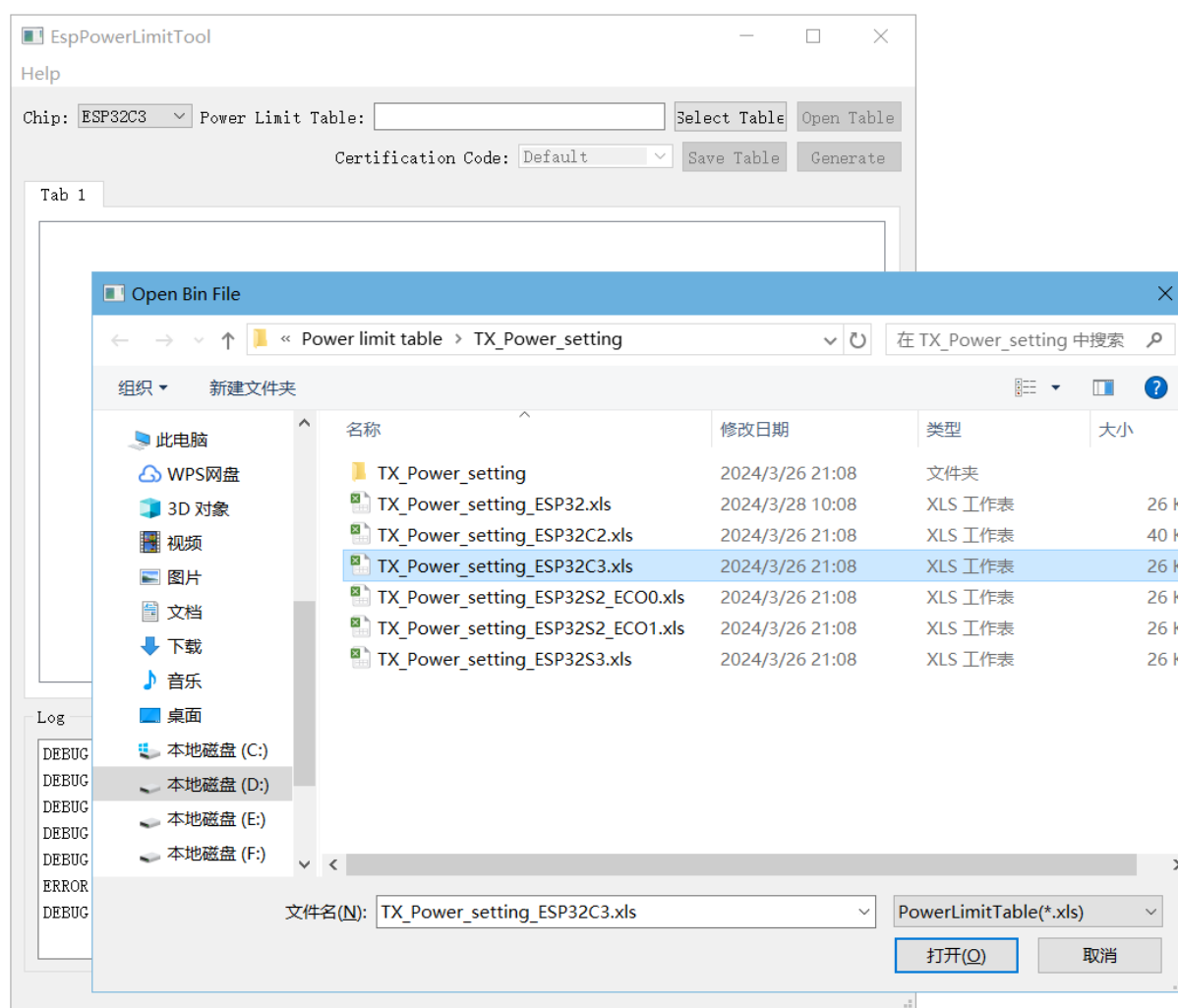


Fig. 9: Importing TX Power Setting Table

3. Click Open Table, modify the power value in the corresponding country code table, and select the desired country code in the Certification Code dropdown box.

1. Fill in the power value based on the certification result (the certification provides the power attenuation value) (Power value = Target power - Attenuation value/4).
2. If Actual_Result is modified, the Target power in the above formula needs to be changed to Actual_Result.
3. Adding or deleting table content is not allowed. For example, FCC only supports channels 1~11, so it is recommended to keep the power values of channels 12~13 in this table the same as channel 11 instead of deleting them;
4. Except for low and high channels, the power values of other channels should be set to the same as the middle channel;
5. The NA section cannot be modified. If the Certification Code cannot be selected from the dropdown box, it indicates that the table has been modified and needs to be restored.

4. Click Save Table to save the settings. Select the required certification from the Certification Code dropdown, then click Generate to create the phy_init_bin file for the corresponding country code.

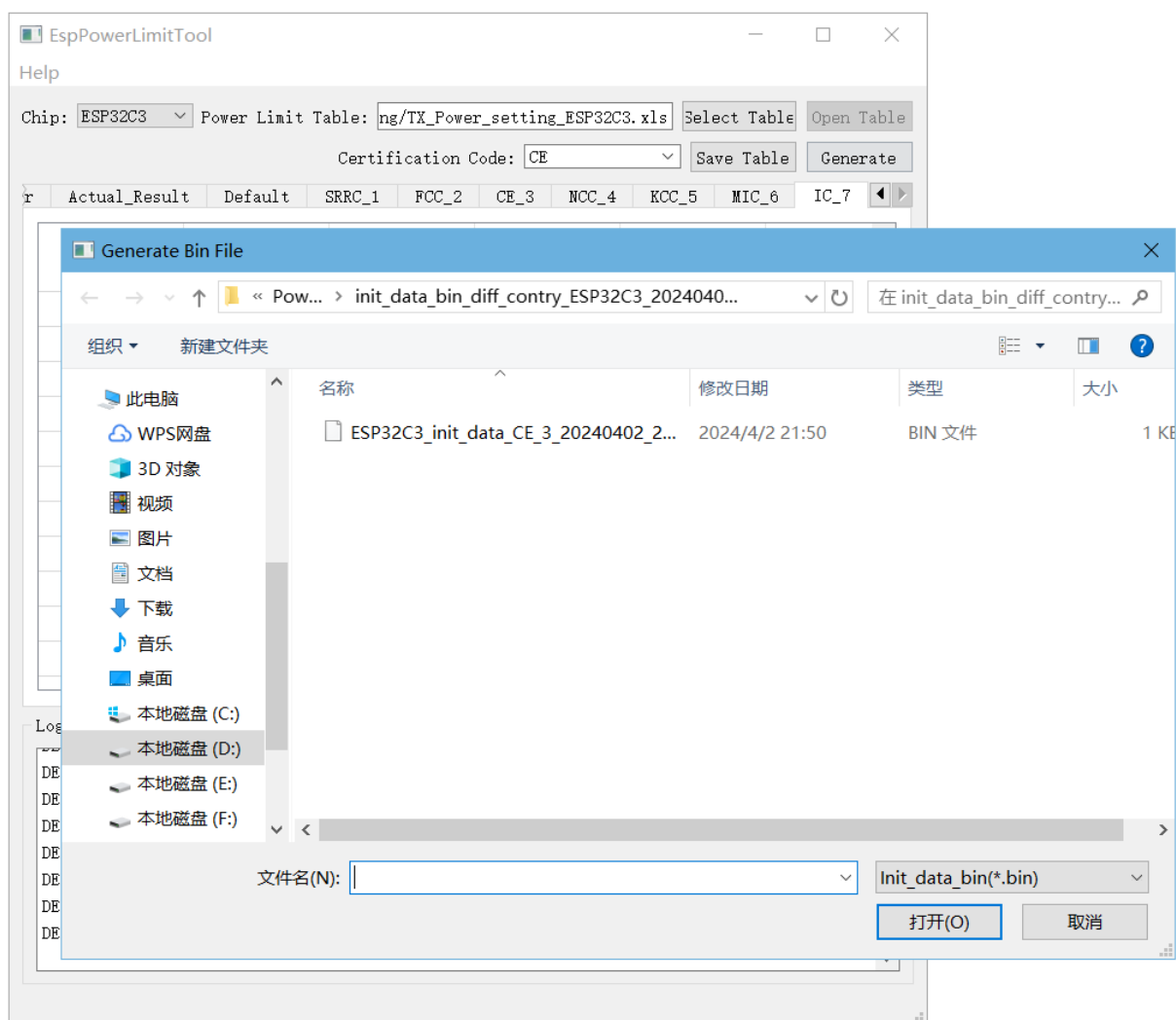


Fig. 11: Generate phy_init_bin File

Note:

1. The dropdown list of *Certification Code* includes options for a single certification, Multiple Country, and Custom.
2. Selecting a single certification will generate a single phy_init_bin file for that certification, which contains a total of 128 bytes except the verification control information.

3. Selecting `Multiple Country` will generate Combined `phy_init_bin` files, including a Default bin file and seven others for SRRC, FCC, CE, NCC, KCC, MIC, and IC. The combined files contain 8*128 bytes.
4. Selecting `Custom` will generate a single or multiple certification bin files based on your choice.

5. Verify whether `phy_init_bin` is effective using Non-Signaling or Signaling Test. Taking Non-Signaling Test as an example, first use the [DownloadTool](#) to download the generated `phy_init_bin` file to the testing product.

- Select `DownloadTool` from `Tool` dropdown list to enter the `DownloadTool` interface.
- Flash the `phy_init_bin` file and corresponding RF test firmware to `flash` by referring to the instructions stated [DownloadTool](#).
- The flash address for `phy_init_bin` is `0x1fc000` and the flash address for the RF test firmware Contact Espressif for the latest firmware is `0x0`.

Note: Regarding the Signaling Test, you can simply replace the original `phy_init_bin`. Please refer to the relevant documents in [RF Test Items](#).

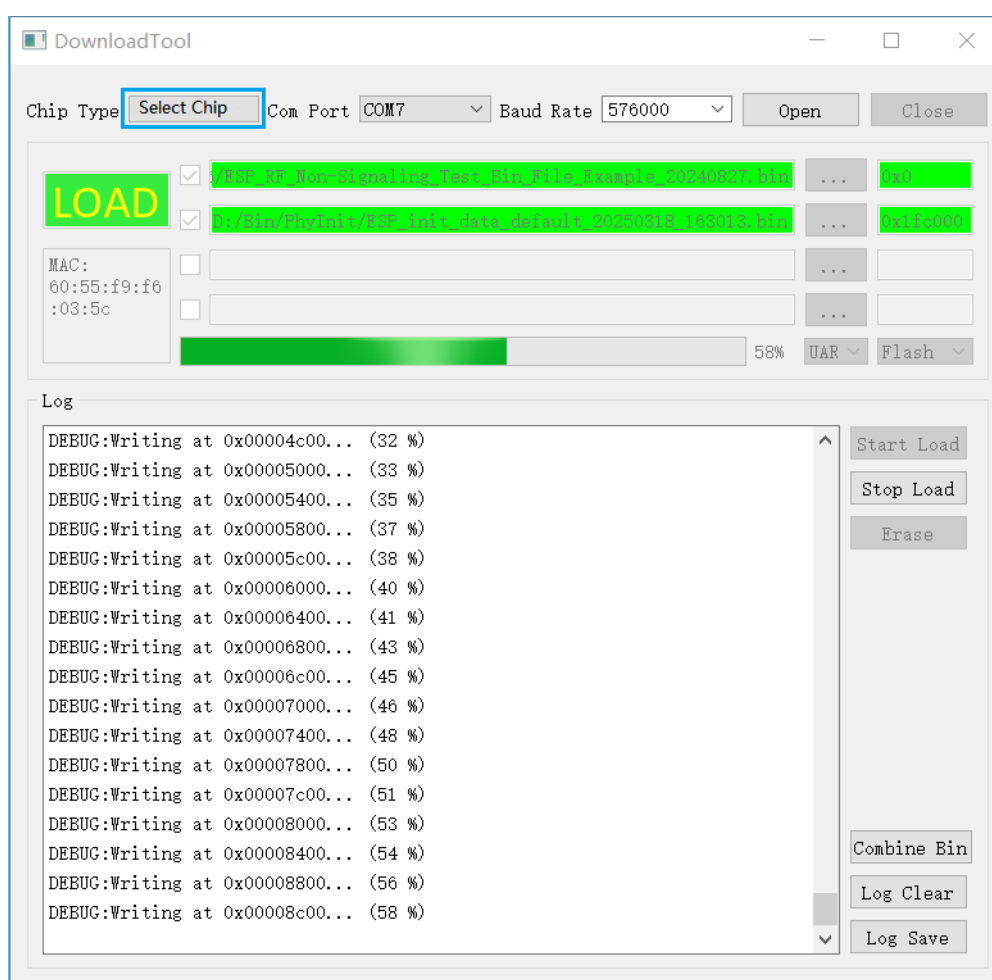


Fig. 12: Flash `phy_init_bin` File

6. Use a Wi-Fi tester to measure the output power and check whether `phy_init_bin` is effective.
 - Open [EspRFTTestTool](#).
 - Select corresponding `ChipType`, `COM`, `BaudRate`, and click `Open` to open the serial port.
 - Open the `WiFi Test` tab, and select `Test Mode`, `Rate`, `BandWidth` and `Channel`.
 - Set `Attenuation` to 0, and `Duty Cycle` to 10%.
 - With `Certification EN` unchecked, i.e., `Phy init` not enabled, the tool tests the initial performance of modules.

- With `Certification EN` checked, i.e., Phy init enabled, the tool tests the performance for certification.
- The default address for flashing `phy_init_bin` is `0x1fc000`. If the flashing address changes, update it here.
- For Multiple Country, you can select the certification country codes it includes in the `Certification Code`.

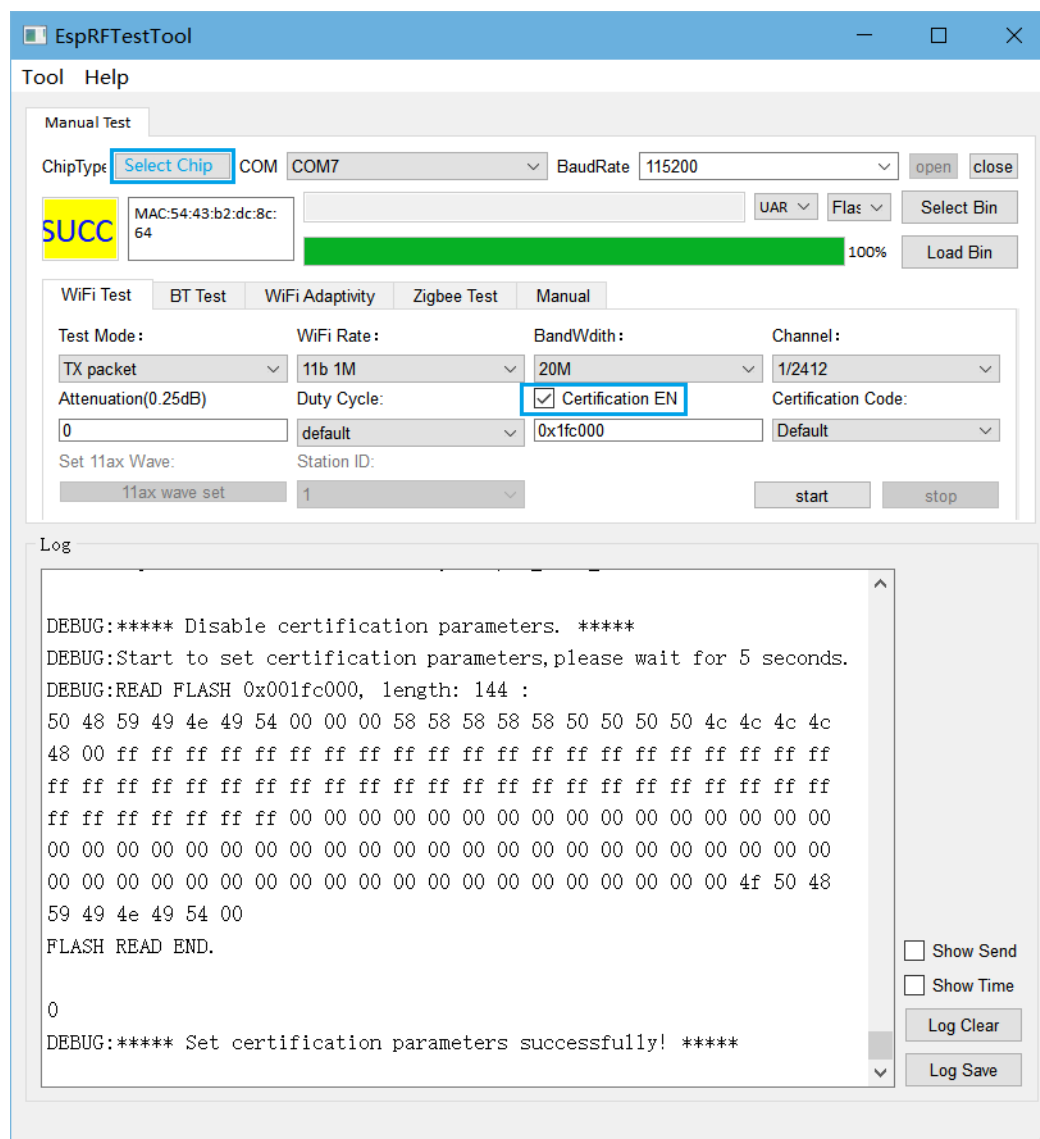


Fig. 13: RF Test Configuration

Typical Average Output Power of ESP32-C61

Rates	Typical Average Output Power (dBm)
11b 1M	20.5
11b 11M	20.5
11g 6M	20
11g 54M	19
HT20-11n MCS0	19
HT20-11n MCS7	18
HT40-11n MCS0	18.5
HT40-11n MCS7	17.5
HE20-11ax MCS0	19
HE20-11ax MCS7	18
HE20-11ax MCS9	15.5

3 RF Test Items

3.1 Wi-Fi Non-Signaling Test

The Wi-Fi Non-Signaling Test, also known as fixed frequency test, directly controls the device to transmit specific signals without establishing a data connection. It evaluates key RF performance metrics, such as transmit power, spectrum quality, and error rate, ensuring wireless communication quality in various scenarios.

Set Up Test Environment

The RF non-signaling test firmware environment mainly includes a PC, tester, a USB-to-UART board, a device under test (DUT), and a shield box.

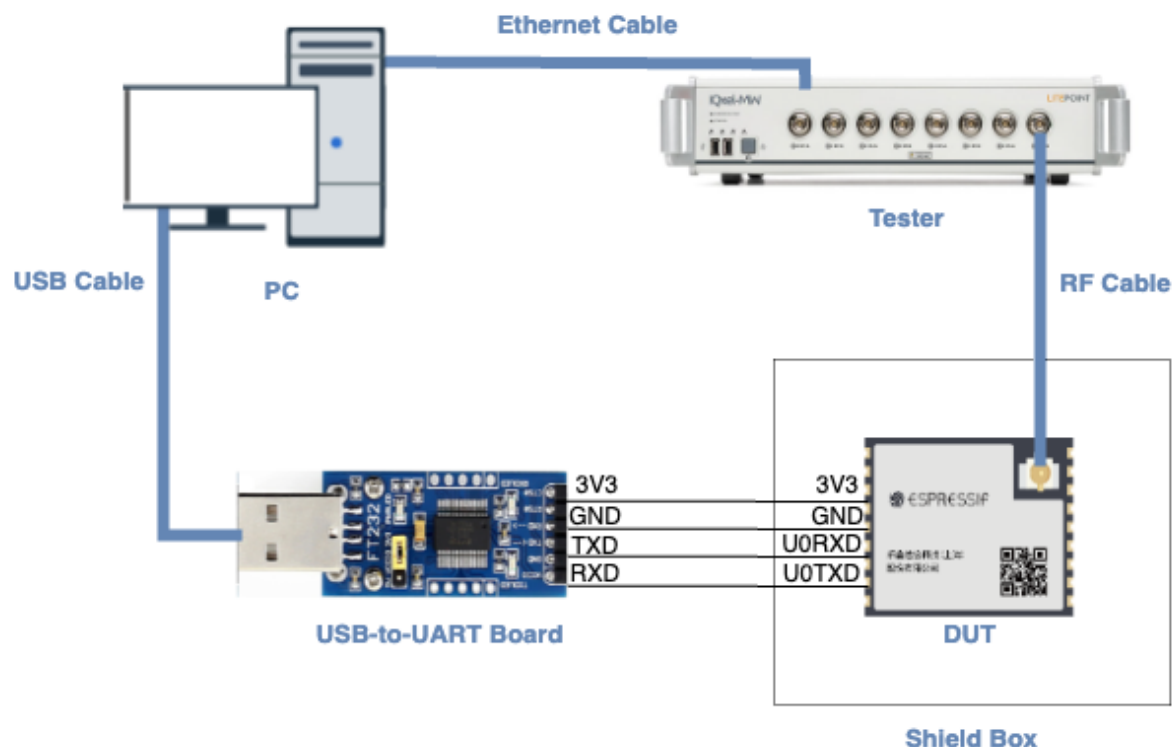


Fig. 14: Test Environment Setup

- **PC** is connected to the USB-to-UART board via USB and to the tester via an Ethernet cable. The PC needs to have the EspRFTTestTool toolkit, tester control software, and the driver for the USB-to-UART board installed.
- **Tester** is used to test the RF performance of the DUT in different modes. Typically, it is the WT-328/IQXel tester.
- **USB-to-UART board** is used to communicate between the PC and the DUT.
- **Device under test (DUT)** refers to a product designed based on the ESP32-C61 chip or module. It is connected to the USB-to-UART board via UART and to the tester via an RF connection cable. The DUT is usually placed inside a shield box.
- **Shield Box** is used to isolate external RF interference and ensure the stability of the test environment.

Note:

- The CHIP_EN pin of the DUT is pulled up by default. If it is not pulled up in the product design, you need to manually connect the CHIP_EN to the 3V3 pin.
- Some serial communication boards have already swapped RXD and TXD internally, so there is no need to reverse the connection. Adjust the wiring according to the actual situation.
- ESP32-C61 has a power-on self-calibration feature. The RF connection cable must be connected to the tester before the DUT is powered on for testing.

Conduction Test

- For modules without an onboard PCB antenna, the RF connection cable can be directly soldered to the antenna feed point of the module (as shown in the schematic diagram above).
- For modules with an onboard PCB antenna, cut the trace that connects to the PCB antenna feed point and solder the RF connection cable. The RF cable's shielding metal layer must be thoroughly soldered before connecting to the module's GND. The GND soldering point can be either the shield cover or the exposed GND layer on the PCB (after removing the green solder mask). Besides, it should be as close to the feed point as possible.



Fig. 15: Soldering RF Connection Cable to Module with Onboard PCB Antenna

Flash Firmware

1. Open [EspRFTestTool](#).
2. Set ChipType, COM, BaudRate, and click Open to open the COM port.

Note: Set BaudRate to 115200

3. Flash Contact Espressif for the latest firmware to Flash via UART.

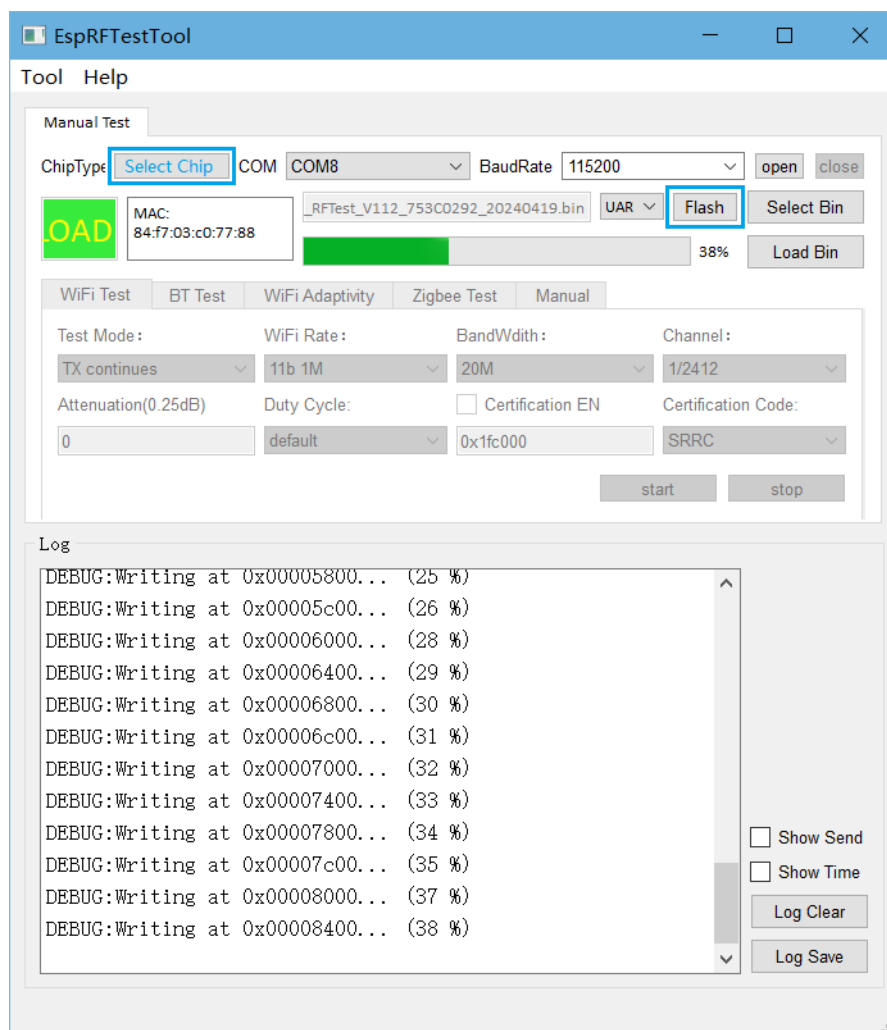


Fig. 16: ESPRFTTestTool Configuration

4. After the firmware flashing is completed, pull the boot pin high or leave it floating. The chip will enter the working mode after power-off restart.

Note: If you use the flash download tool to flash the firmware, change the flash address of ESP32-C61 to 0x0.

Start Testing

Wi-Fi TX Performance Test

- **Test Mode:**
 - TX packet: Packet transmission duty cycle less than 50%, used for TX performance tests
 - TX continues: Packet duty cycle close to 100%, used for certification tests;
 - TX tone: Used for single-carrier tests.
- **Wi-Fi Rate:** Set Wi-Fi test rate
- **BandWidth:** Set Wi-Fi test bandwidth
- **Channel:** Set Wi-Fi test channel
- **Attenuation (0.25 dB):** Set power attenuation
 - 0 means no attenuation, which is the default value;
 - 2 means 0.5 dB attenuation;
 - 4 means 1 dB attenuation, and so on.

- **Duty Cycle:** Set the packet duty cycle in TX packet tests. The default duty cycle is around 30%. Supported values: 10%, 50%, 90%.
- **Certification EN:** Not enabled by default. Used only when verifying Power Limit function.
- **Certification Code:** Not enabled by default. Used only when verifying Power Limit function.

After clicking **start**, the log window should print Wi-Fi transmission parameters similar to the following:

```
Wifi tx out: channel=1, rate=0x0, BK=0, length=50, delay=1200, packet_num=0
```

The above parameters indicate that Wi-Fi packet transmission is normal, and the transmission performance can be detected with tester at this time.

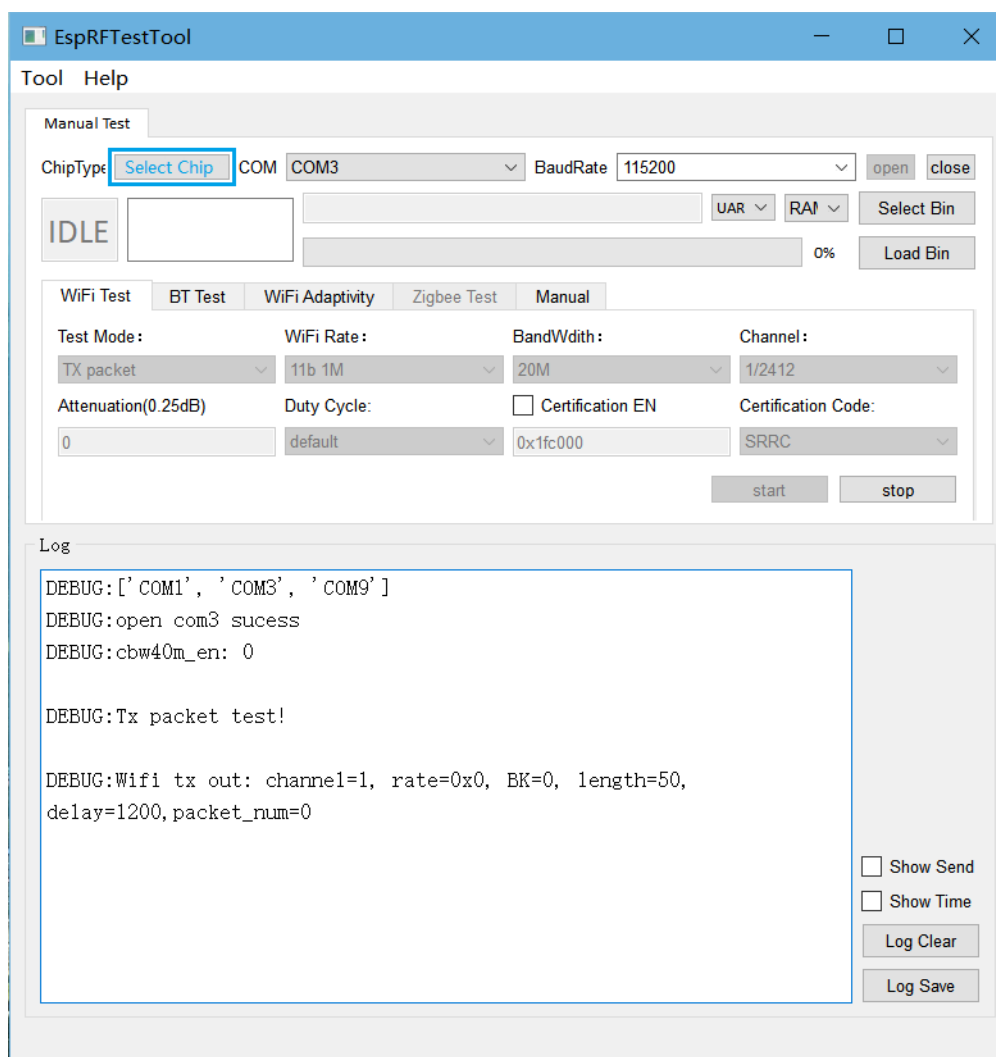


Fig. 17: Wi-Fi TX Performance Test

Wi-Fi RX Performance Test

- **Test Mode:** Set to RX packet for RX performance tests.
- **Wi-Fi Rate:** Set Wi-Fi test rate.
- **BandWidth:** Set Wi-Fi test bandwidth.
- **Channel:** Set Wi-Fi test channel.

After clicking **start**, the tester sends packets on the test channel. Click **stop** after completion. The log window should display packet RX information similar to the following:

```
Correct:1000 Desired:1000 RSSI:-614 noise:-960 gain:0 paral:0 para2:0 freq:0
```

Where:

- **Correct:** The total number of packets received this time.
- **Desired:** The number of packets received at the corresponding rate this time.
- **RSSI:** Represents the average RSSI of the received Desired packets. For example, “RSSI: -614” means the RSSI value is -61.4.

Note:

- If Desired is 0, no packets were received from the tester. Please check the tester's packet settings and packet file to ensure the packet RX link is normal;
- If Desired is not 0 and Correct is greater than Desired, there is interference in the environment. Please retest in a shielded environment;
- Other parameters in the packet RX information are only used for RD debug and have no actual meaning.

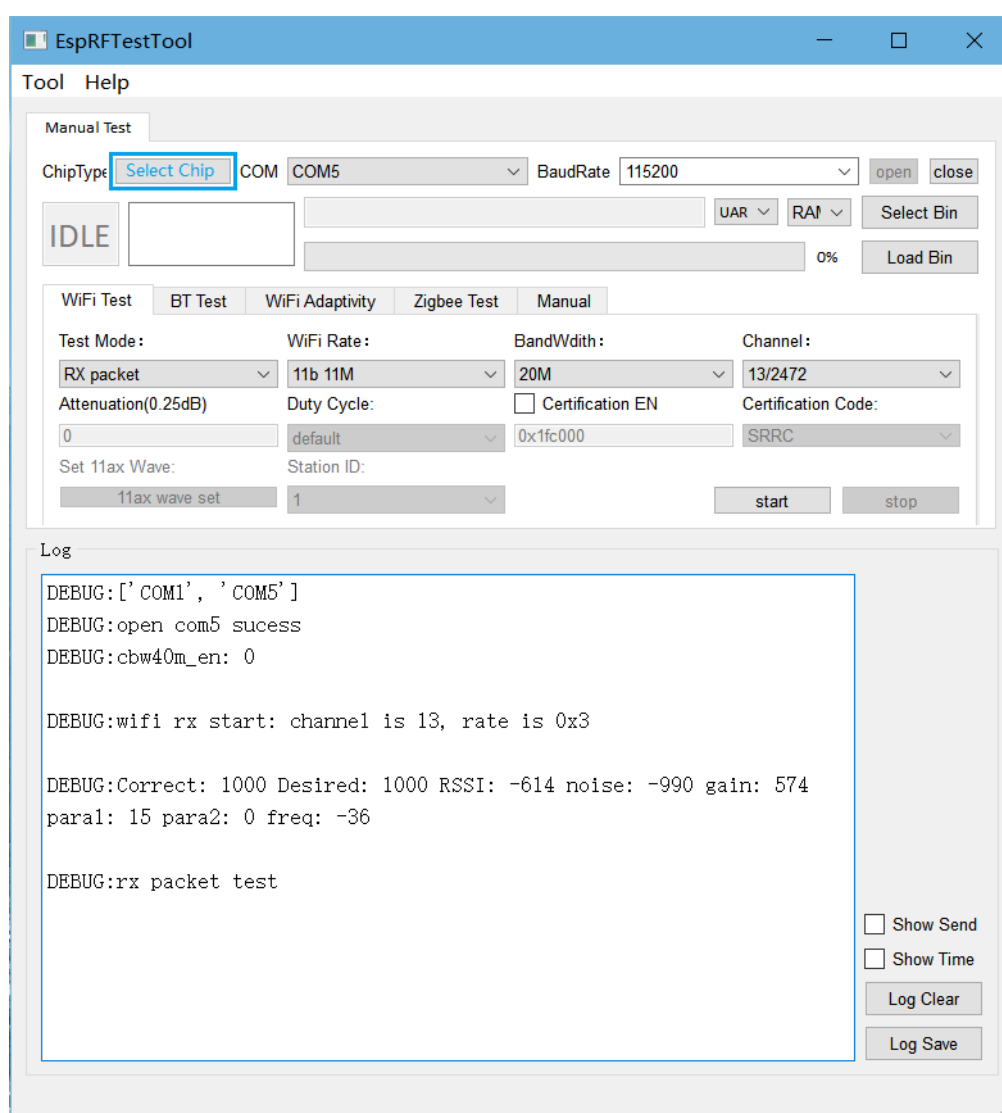


Fig. 18: Wi-Fi RX Performance Test

Appendix

This appendix is mainly used to explain the target output power of the chip's Wi-Fi, which is used for RF debugging or test reference.

Table 2: ESP32-C61 Wi-Fi Target TX Power

Rate	ESP32-C61 Wi-Fi Target Power (dBm)
11b 1M	20.5
11b 11M	20.5
11g 6M	20
11g 54M	19
HT20-11n MCS0	19
HT20-11n MCS7	18
HT40-11n MCS0	18.5
HT40-11n MCS7	17.5
HE20-11ax MCS0	19
HE20-11ax MCS7	18
HE20-11ax MCS9	15.5

3.2 Wi-Fi Signaling Test

The Wi-Fi Signaling Test assesses and verifies the Wi-Fi signaling functions of wireless network devices, focusing on stable and reliable communication across varying operating scenarios. It evaluates Over-The-Air (OTA) performance, including Total Radiated Power (TRP) and Total Isotropic Sensitivity (TIS).

Set Up Test Environment

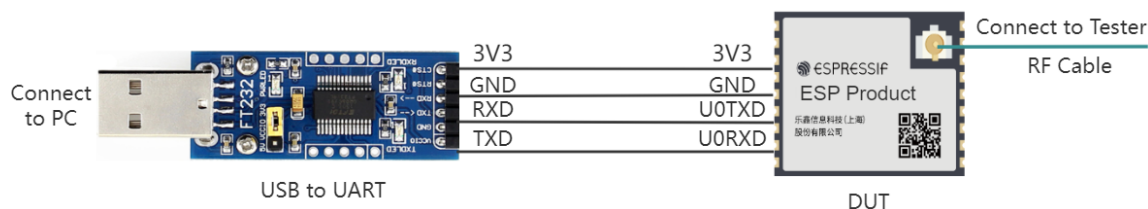


Fig. 19: UART Connection Description

The **Device Under Test (DUT)** is a product designed based on Espressif chips or modules. The device under test is connected to the USB-to-UART adapter board via UART.

Note:

- The CHIP_EN pin of the device under test is pulled up by default. If it is not pulled up in the product design, you need to manually connect the CHIP_EN to the 3V3 pin.
- Some serial communication boards have already swapped RXD and TXD internally, so there is no need to reverse them. The wiring needs to be adjusted according to the actual situation.
- Espressif chips have a power-on self-calibration feature. Therefore, before powering on the device under test, the RF connection cable must be connected to the testing instrument.

Flash Firmware

1. Open [DownloadTool](#).
2. Set ChipType, Com Port and Baud Rate, click Open, and select download to Flash.
3. Contact Espressif for the latest firmware supports a single country code, Contact Espressif for the latest firmware supports multiple country codes. They each include 4 bin files, i.e., **bootloader.bin**, **partition-table.bin**, **phy_init_data.bin**, and **ssc.bin**.

After unzipping Contact Espressif for the latest firmware or Contact Espressif for the latest firmware, flash the 4 bin files to the following addresses via UART.

bin file	flashing address
bootloader.bin	0x0
partition-table.bin	0x8000
phy_init_data.bin	0xF000
ssc.bin	0x10000

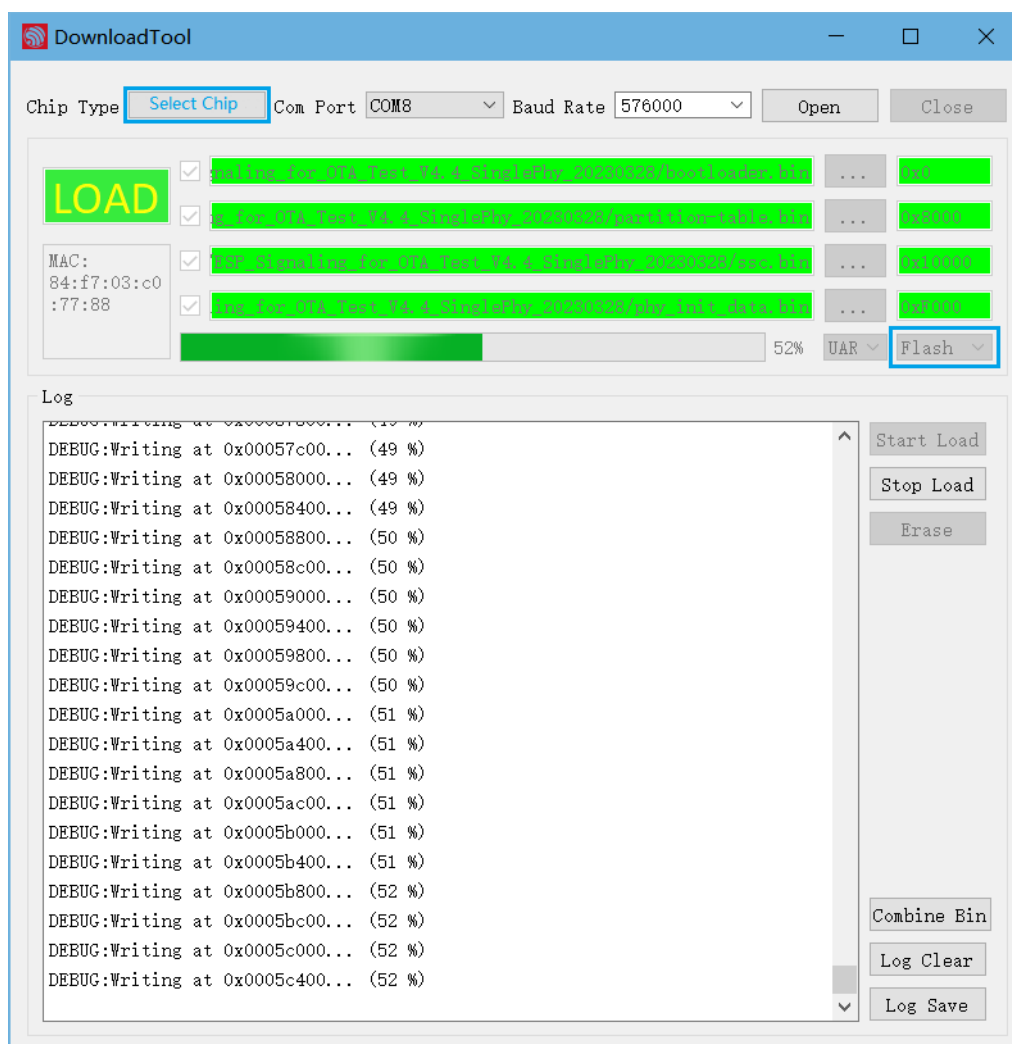


Fig. 20: Firmware Flashing Schematic

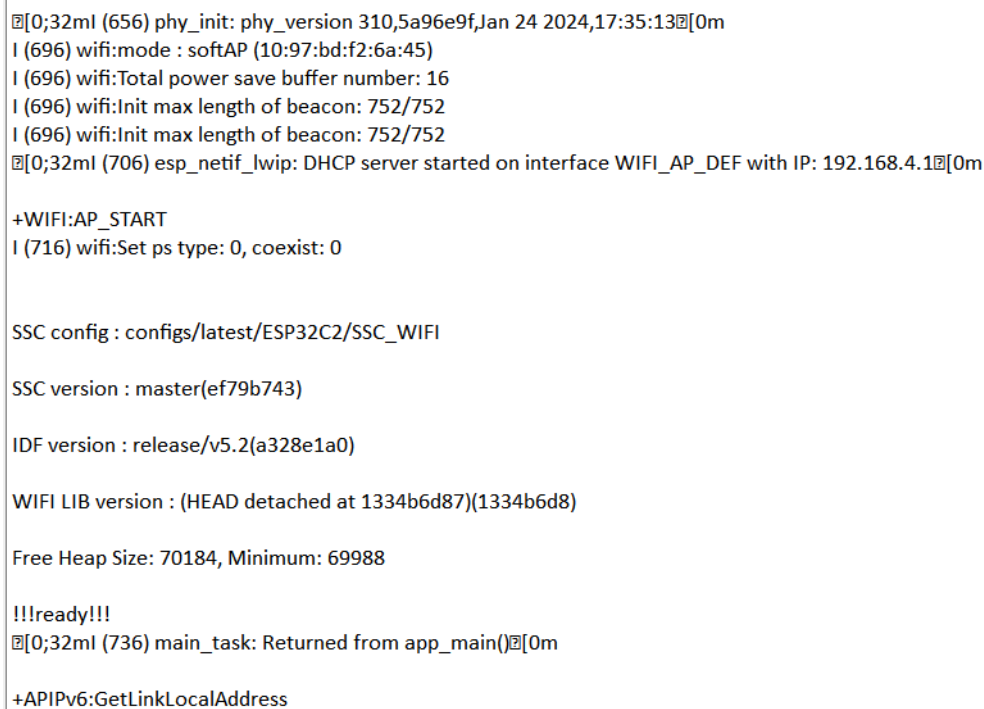
After the flashing is completed, continue with the following steps for signaling testing.

Start Testing

Check Power-on Log

Note: BaudRate is set to 115200.

Use a serial communication tool, such as [Friendly Serial Assistant](#), to configure the port number and baud rate. After the device under test is powered on again, if the serial output is similar to the following information, you can confirm that the test status is normal:



```

[0;32mI (656) phy_init: phy_version 310,5a96e9f,Jan 24 2024,17:35:13[0m
I (696) wifi:mode : softAP (10:97:bd:f2:6a:45)
I (696) wifi:Total power save buffer number: 16
I (696) wifi:Init max length of beacon: 752/752
I (696) wifi:Init max length of beacon: 752/752
[0;32mI (706) esp_netif_lwip: DHCP server started on interface WIFI_AP_DEF with IP: 192.168.4.1[0m

+WIFI:AP_START
I (716) wifi:Set ps type: 0, coexist: 0

SSC config : configs/latest/ESP32C2/SSC_WIFI

SSC version : master(ef79b743)

IDF version : release/v5.2(a328e1a0)

WIFI LIB version : (HEAD detached at 1334b6d87)(1334b6d8)

Free Heap Size: 70184, Minimum: 69988

!!!ready!!!
[0;32mI (736) main_task: Returned from app_main()[0m

+APIv6:GetLinkLocalAddress
  
```

Fig. 21: Serial Port Log for Device Power-on

Device Provisioning Enter the following two commands in the serial port in sequence for network configuration.

```

//Device Provisioning
//Configure the prototype to enter station mode
op -S -o 1

//Connect to AP, SSID is CMW-AP, password is 12345678
sta -C -s CMW-AP -p 12345678
  
```

Note: The `-p` parameter is used to set the AP password. If the AP has no password, this parameter is not needed.

After the station device is assigned an IP address, the Wi-Fi connection is successful, and the following log is printed:

```

I (325546) wifi:new:<1,0>, old:<1,0>, ap:<1,0>, sta:<255,255>, prof:1

+SOFTAP:STADISCONNECTED,42:37:dd:d6:40:44,3
op -S -o 1
I (1407226) wifi:mode : sta (10:97:bd:f2:6a:44)
I (1407226) wifi:enable tsf

+WIFI:AP_STOP

+MODE:OK

+WIFI:STA_START
sta -C -s CMW-AP1 -p 12345678

+JAP:OK
I (1709076) wifi:new:<6,0>, old:<1,0>, ap:<255,255>, sta:<6,0>, prof:1
I (1709356) wifi:state: init -> auth (b0)
I (1709366) wifi:state: auth -> assoc (0)
I (1709366) wifi:state: assoc -> run (10)
I (1709376) wifi:connected with CMW-AP1, aid = 1, channel 6, BW20, bssid = c8:0e:77:4f:d4:29
I (1709376) wifi:security: WPA2-PSK, phy: bgn, rssi: -41
I (1709396) wifi:pm start, type: 0

I (1709396) wifi:dp: 1, bi: 102400, li: 3, scale listen interval from 307200 us to 307200 us
I (1709396) wifi:set rx beacon pti, rx_bcn_pti: 0, bcn_timeout: 25000, mt_pti: 0, mt_time: 10000

+JAP:WIFICONNECTED
I (1709436) wifi:AP's beacon interval = 102400 us, DTIM period = 1

+STAIPv6:GetLinkLocalAddress
[0;32mI (1712406) esp_netif_handlers: sta ip: 192.168.5.8, mask: 255.255.255.0, gw: 192.168.5.1[0m

+JAP:CONNECTED,CMW-AP1

```

Fig. 22: Serial Port Log for Device Provisioning

After the device under test is successfully connected, you can use the RF test instrument for Wi-Fi Signaling Test.

3.3 Wi-Fi Adaptivity Test

The Wi-Fi Adaptivity Test evaluates a device's ability to make real-time adjustments to parameters, such as transmission rate, channel selection, and power levels, by simulating varying network conditions and loads. This test aims to optimize the overall network performance and stability.

Note: If the power spectral density (PSD) of the Wi-Fi signal is higher than 10 dBm/MHz, the adaptivity test should choose the Listen Before Talk (LBT) mechanism based on non-hopping load.

Set Up Test Environment

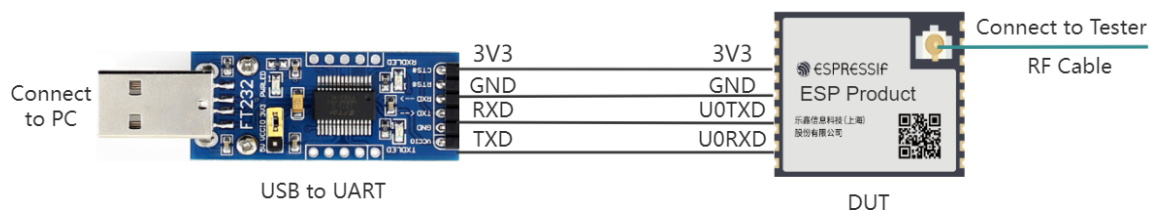


Fig. 23: UART Connection Description

The **Device Under Test (DUT)** is a product designed based on Espressif chips or modules. The DUT is connected to the USB-to-UART adapter board via UART.

Note:

- The CHIP_EN pin of the DUT is pulled up by default. If it is not pulled up in the product design, you need to manually connect the CHIP_EN to the 3V3 pin.
- Some serial communication boards have already swapped RXD and TXD internally, so there is no need to reverse them. The wiring should be adjusted according to the actual situation.
- Espressif chips have a power-on self-calibration function, so the RF connection line must be connected to the test instrument before the DUT is powered on for testing.

Flash Firmware

1. Open [DownloadTool](#).
2. Set ChipType, Com Port, Baud Rate, click Open, select to download to Flash.
3. Flash Contact Espressif for the latest firmware to 0x0 via UART.

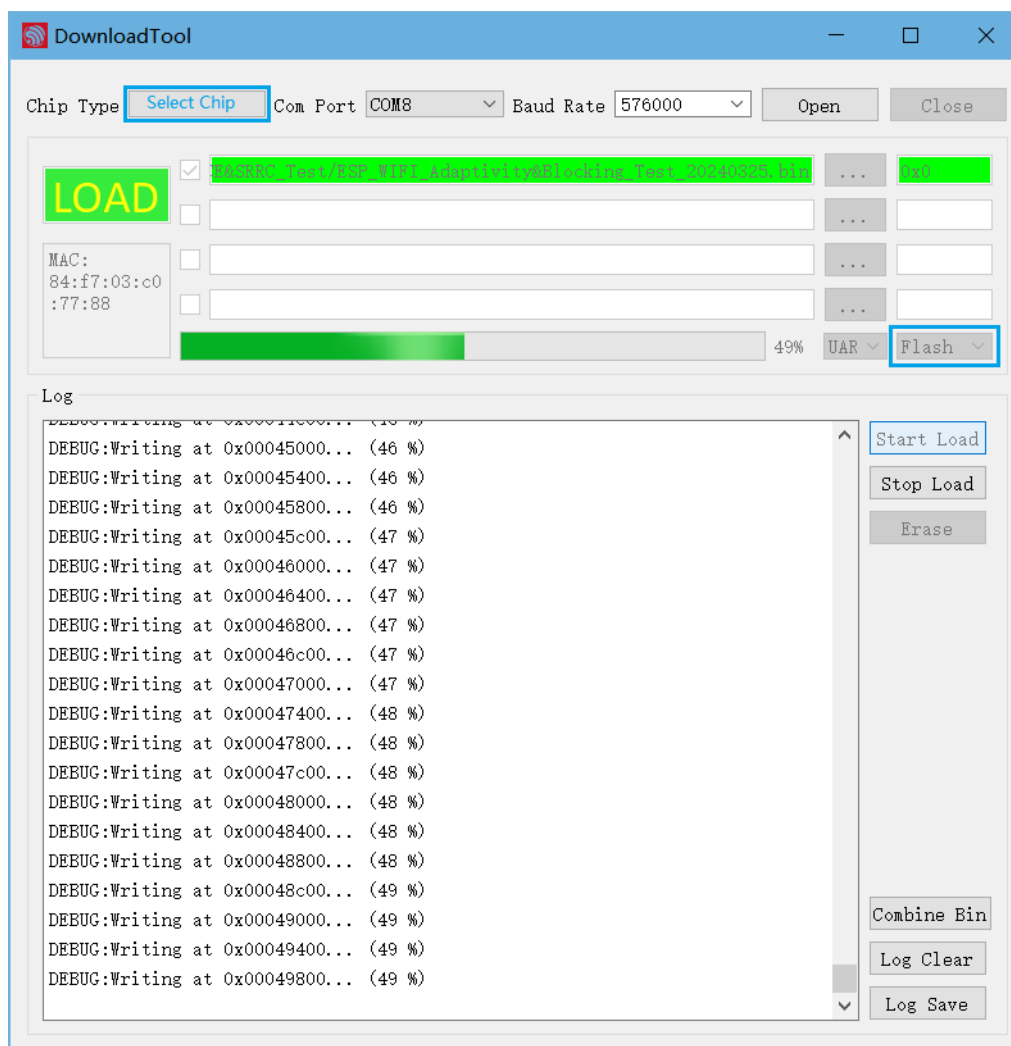


Fig. 24: Flashing Firmware

After the flashing is completed, continue with the following steps for the adaptivity test.

Start Testing

Note: Set BaudRate to 115200.

Check Power-on log Use a serial communication tool, such as [SerialPortUtility](#), and configure the port number and the baud rate. If the device under test prints similar information as below after being powered on again, you can confirm that the test status is OK:

```

[0;32mI (656) phy_init: phy_version 310,5a96e9f,Jan 24 2024,17:35:13[0m
I (696) wifi:mode : softAP (10:97:bd:f2:6a:45)
I (696) wifi:Total power save buffer number: 16
I (696) wifi:Init max length of beacon: 752/752
I (696) wifi:Init max length of beacon: 752/752
[0;32mI (706) esp_netif_lwip: DHCP server started on interface WIFI_AP_DEF with IP: 192.168.4.1[0m

+WIFI:AP_START
I (716) wifi:Set ps type: 0, coexist: 0

SSC config : configs/latest/ESP32C2/SSC_WIFI

SSC version : master(ef79b743)

IDF version : release/v5.2(a328e1a0)

WIFI LIB version : (HEAD detached at 1334b6d87)(1334b6d8)

Free Heap Size: 70184, Minimum: 69988

!!!ready!!!
[0;32mI (736) main_task: Returned from app_main()[0m

+APIv6:GetLinkLocalAddress

```

Fig. 25: Device Power-on Serial Port Print Log

Next, you can choose to [test using serial port commands](#) or [test using EspRFTestTool tool](#).

Test Using Serial Port Commands Enter the following commands in the serial port in sequence for network configuration and traffic testing:

```

//Device provisioning
//Configure the prototype to enter station mode
op -S -o 1

//Connect to AP, SSID is CMW-AP, password is 12345678
sta -C -s CMW-AP -p 12345678

//Traffic test
//Clear socket
soc -T

//Create UDP, port is 8080, default socket ID is 54
soc -B -t UDP -p 8080

//Perform traffic test on AP device with socket ID 54
soc -S -s 54 -i 192.168.1.1 -p 8080 -l 2000 -n 200000000 -j 1

```

Note: The `-p` parameter is used to set the AP password. If the AP has no password, this parameter is not needed.

If the following similar information is printed in the serial port, it indicates that the traffic has been started and the Wi-Fi Adaptivity Test can be initiated.

```

+APIv6:GetLinkLocalAddress
op -S -o 1
I (4041) wifi:mode : sta (10:97:bd:f2:6a:44)
I (4041) wifi:enable tsf

+WIFI:AP_STOP

+MODE:OK

+WIFI:STA_START
sta -C -s CMW-AP1 -p 12345678

+JAP:OK
I (6101) wifi:new:<6,0>, old:<1,0>, ap:<255,255>, sta:<6,0>, prof:1
I (6381) wifi:state: init -> auth (b0)
I (6381) wifi:state: auth -> assoc (0)
I (6391) wifi:state: assoc -> run (10)
I (6421) wifi:connected with CMW-AP1, aid = 1, channel 6, BW20, bssid = c8:0e:77:4f:d4:29
I (6421) wifi:security: WPA2-PSK, phy: bgn, rssi: -39
I (6431) wifi:pm start, type: 0

I (6431) wifi:dp: 1, bi: 102400, li: 3, scale listen interval from 307200 us to 307200 us
I (6431) wifi:set rx beacon pti, rx_bcn_pti: 0, bcn_timeout: 25000, mt_pti: 0, mt_time: 10000

+JAP:WIFICONNECTED
I (6511) wifi:AP's beacon interval = 102400 us, DTIM period = 1
[0;32mI (7441) esp_netif_handlers: sta ip: 192.168.67.174, mask: 255.255.255.0, gw: 192.168.67.1[0m

+JAP:CONNECTED,CMW-AP1

+STAIPv6:GetLinkLocalAddress
soc -T

+CLOSEALL

soc -B -t UDP -p 8080

+BIND:54,OK,0.0.0.0,8080
soc -S -s 54 -i 192.168.1.1 -p 8080 -l 2000 -n 2000000000000 -j 1

```

Fig. 26: Serial Port Log for Device Provisioning

Test Using EspRFTestTool Tool

- Open the EspRFTestTool toolkit, configure ChipType, COM, and BaudRate, open the port, and select the WiFi Adaptivity test interface.
- In STA mode, enter AP ssid and AP pwd, and click Connect AP to connect.
- After successful connection, the following log should be printed:

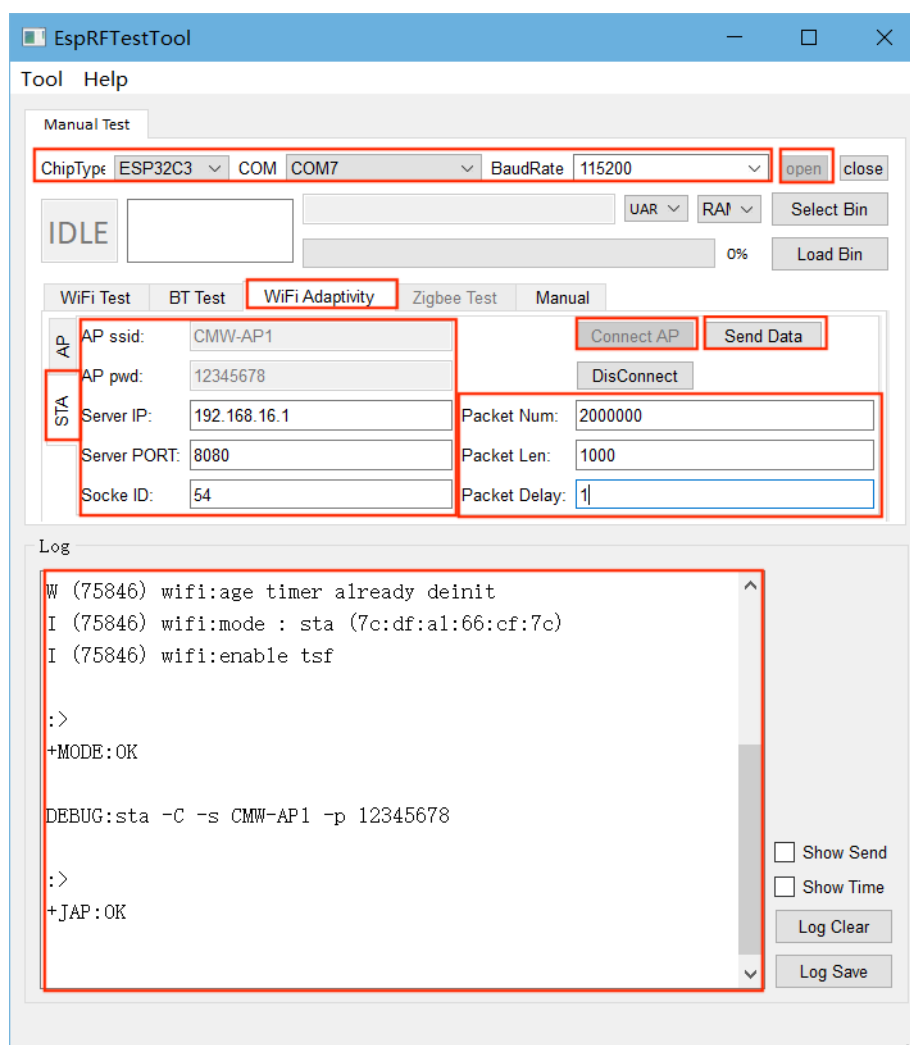


Fig. 27: Device Network Provisioning

- After a successful connection, set `Packet Num` to a sufficiently large value—such as 20000000—to ensure the traffic can run for a long duration.
- Set `Server PORT` to 8080, `Socket ID` to 54, and change `Packet Delay` to 1 to meet certification requirements.
- After the above settings are completed, click `Send Data`. If the log is similar to the figure below, it indicates that the traffic has been started, and the Wi-Fi Adaptivity Test can be initiated.

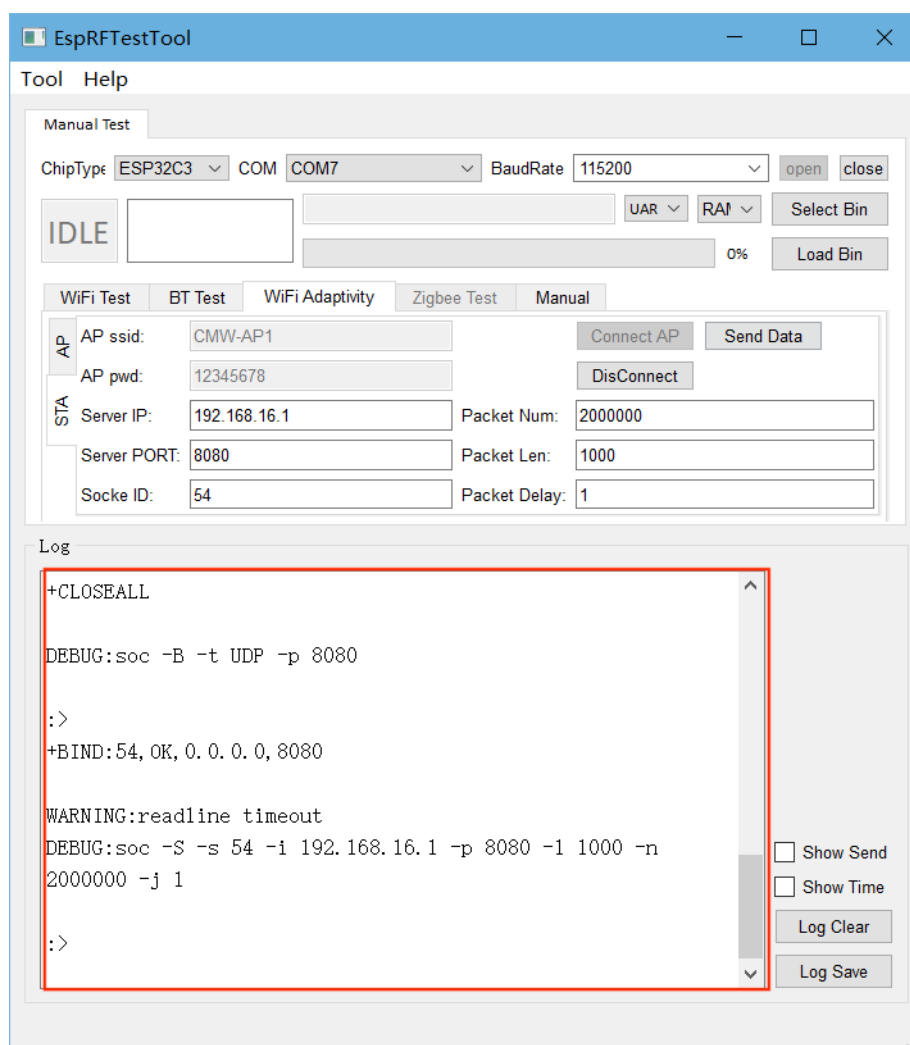


Fig. 28: Wi-Fi Adaptivity Traffic Test

3.4 Wi-Fi Blocking Test

The Wi-Fi Blocking Test evaluates the device's reception performance in environments with strong interference. By introducing high-intensity interference signals, it measures the reception sensitivity and anti-interference capability of a device, ensuring reliable operation in complex wireless environments.

Set Up Test Environment

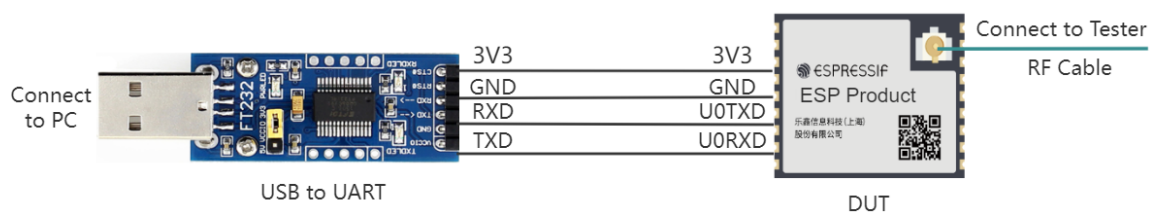


Fig. 29: UART Connection Description

The **Device Under Test (DUT)** is a product designed based on Espressif chips or modules. The DUT is connected to the USB-to-UART adapter board via UART.

Note:

- The CHIP_EN pin of the DUT is pulled up by default. If it is not pulled up in the product design, you need to manually connect the CHIP_EN to the 3V3 pin.
- Some serial communication boards have already swapped RXD and TXD internally, so there is no need to reverse them. The wiring should be adjusted according to the actual situation.
- Espressif chips have a power-on self-calibration function, so the RF connection line must be connected to the test instrument before the DUT is powered on for testing.

Flash Firmware

1. Open [DownloadTool](#).
2. Set ChipType, Com Port, Baud Rate, click Open, select to download to Flash.
3. Flash Contact Espressif for the latest firmware to 0x0 via UART.

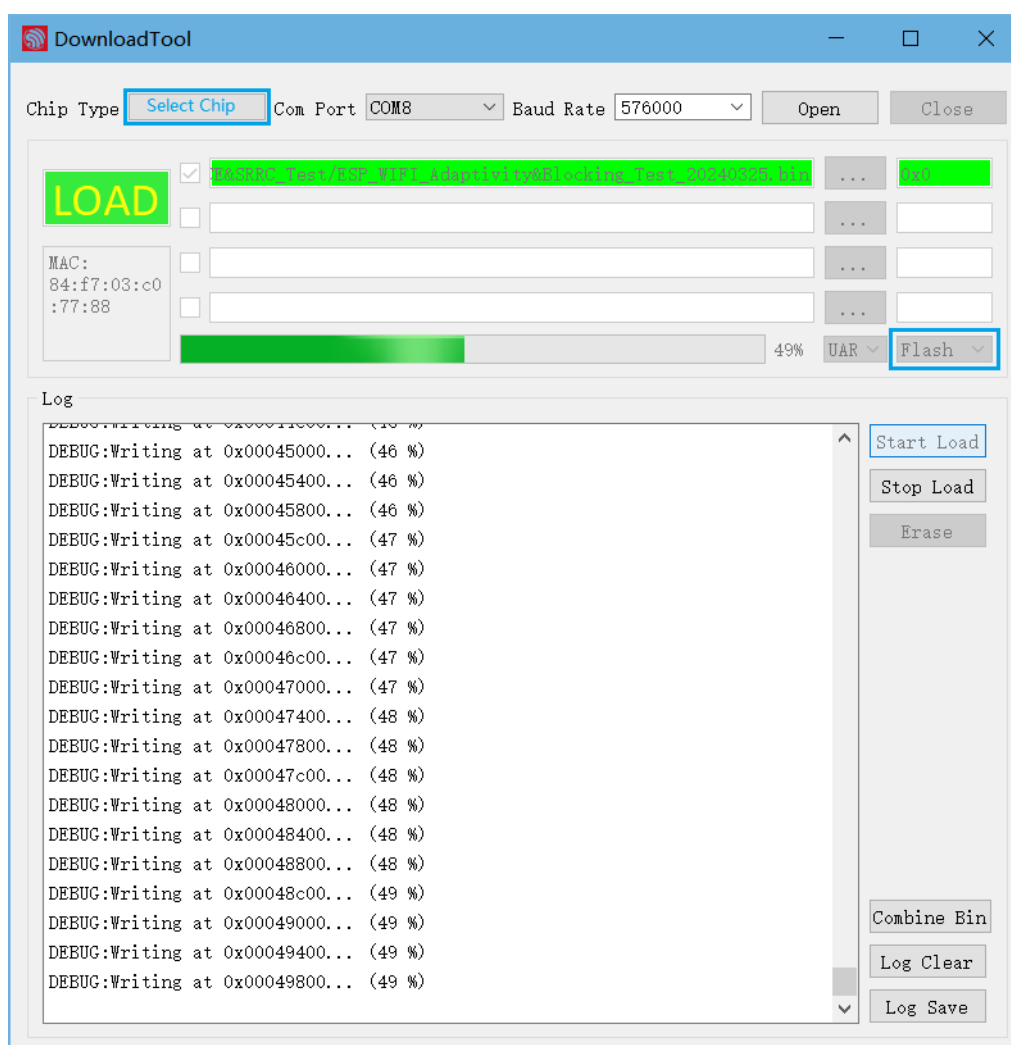


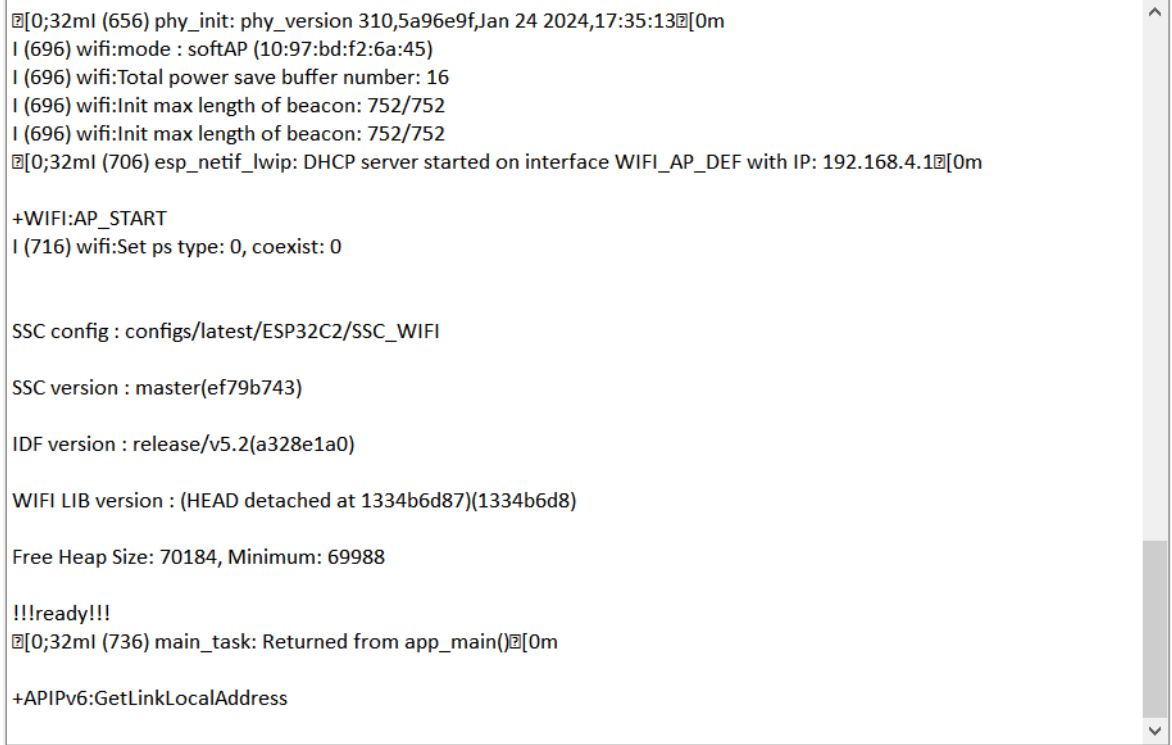
Fig. 30: Flashing Firmware

After the flashing is completed, continue with the following steps for the adaptivity test.

Start Testing

Note: Set BaudRate to 115200.

Check Power-on log Use a serial communication tool, such as [SerialPortUtility](#), and configure the port number and the baud rate. If the device under test prints similar information as below after being powered on again, you can confirm that the test status is OK:



```
[0;32mI (656) phy_init: phy_version 310,5a96e9f,Jan 24 2024,17:35:13[0m
I (696) wifi:mode : softAP (10:97:bd:f2:6a:45)
I (696) wifi:Total power save buffer number: 16
I (696) wifi:Init max length of beacon: 752/752
I (696) wifi:Init max length of beacon: 752/752
[0;32mI (706) esp_netif_lwip: DHCP server started on interface WIFI_AP_DEF with IP: 192.168.4.1[0m

+WIFI:AP_START
I (716) wifi:Set ps type: 0, coexist: 0

SSC config : configs/latest/ESP32C2/SSC_WIFI

SSC version : master(ef79b743)

IDF version : release/v5.2(a328e1a0)

WIFI LIB version : (HEAD detached at 1334b6d87)(1334b6d8)

Free Heap Size: 70184, Minimum: 69988

!!!ready!!!
[0;32mI (736) main_task: Returned from app_main()[0m

+APIv6:GetLinkLocalAddress
```

Fig. 31: Device Power-on Serial Port Print Log

Test with Serial Port Commands Enter the following commands in the serial port in sequence to set up the network:

```
// Device Provisioning
// Configure the prototype to enter station mode
op -S -o 1

// Connect to AP, SSID is CMW-AP, password is 12345678
sta -C -s CMW-AP -p 12345678
```

Note:

- The `-p` parameter is used to set the AP password. If the AP has no password, this parameter is not needed.
-

If the following information is printed on the serial port, the connection is successful and the Wi-Fi Blocking Test can be performed.


```

+APIIPv6:GetLinkLocalAddress
op -S -o 1
I (4041) wifi:mode : sta (10:97:bd:f2:6a:44)
I (4041) wifi:enable tsf

+WIFI:AP_STOP

+MODE:OK

+WIFI:STA_START
sta -C -s CMW-AP1 -p 12345678

+JAP:OK
I (6101) wifi:new:<6,0>, old:<1,0>, ap:<255,255>, sta:<6,0>, prof:1
I (6381) wifi:state: init -> auth (b0)
I (6381) wifi:state: auth -> assoc (0)
I (6391) wifi:state: assoc -> run (10)
I (6421) wifi:connected with CMW-AP1, aid = 1, channel 6, BW20, bssid = c8:0e:77:4f:d4:29
I (6421) wifi:security: WPA2-PSK, phy: bgn, rssi: -39
I (6431) wifi:pm start, type: 0

I (6431) wifi:dp: 1, bi: 102400, li: 3, scale listen interval from 307200 us to 307200 us
I (6431) wifi:set rx beacon pti, rx_bcn_pti: 0, bcn_timeout: 25000, mt_pti: 0, mt_time: 10000

+JAP:WIFICONNECTED
I (6511) wifi:AP's beacon interval = 102400 us, DTIM period = 1
[0;32mI (7441) esp_netif_handlers: sta ip: 192.168.67.174, mask: 255.255.255.0, gw: 192.168.67.1[0m

+JAP:CONNECTED,CMW-AP1

+STAIPv6:GetLinkLocalAddress

```

Fig. 32: Serial Port Log for Device Provisioning

Test with ESPRFTTestTool

- Open the [EspRFTTestTool package](#), configure ChipType and COM, select 115200 for BaudRate, open the port, and select the WiFi Adaptivity test interface.
- In STA mode, enter AP ssid and AP pwd, and click Connect AP to connect.
- After successful connection, the following log should be printed:

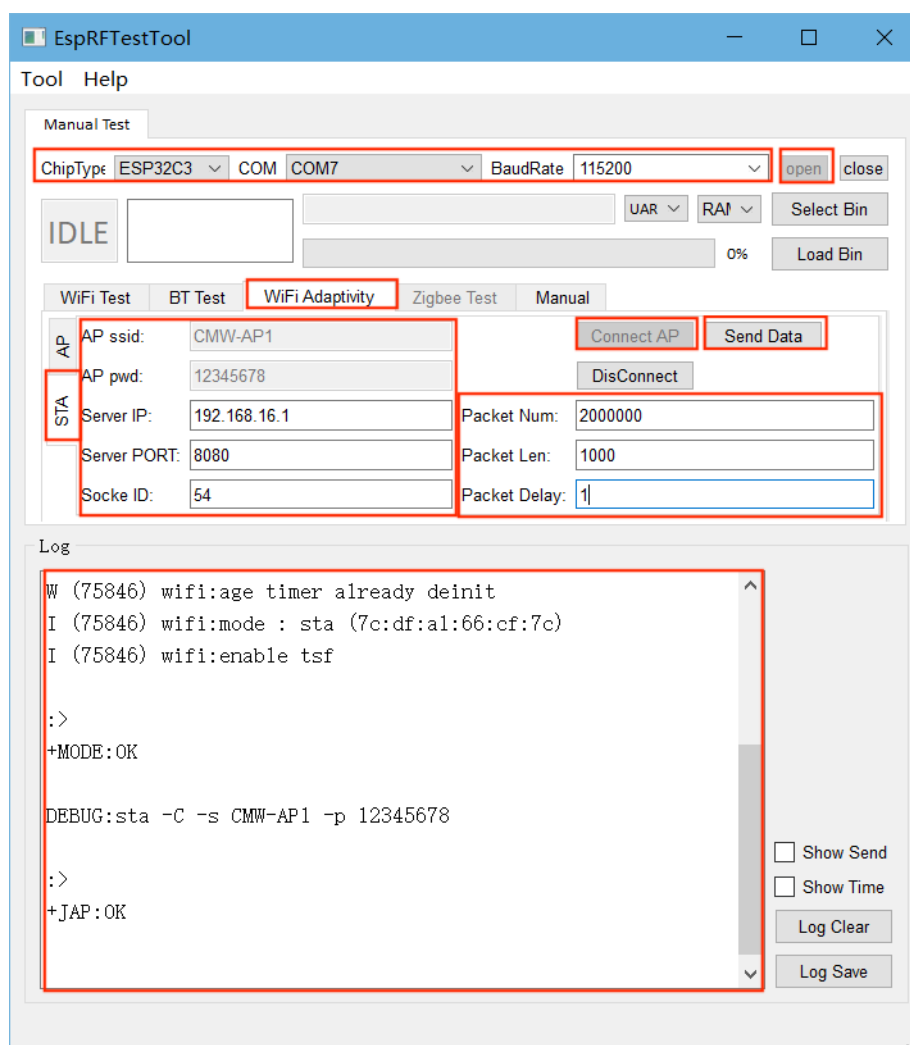


Fig. 33: Serial Port Log for Device Provisioning

After successful connection, you can start the Wi-Fi Blocking Test.

3.5 Bluetooth LE Non-Signaling Test

Bluetooth LE Non-Signaling Test controls the device to transmit specific signals without establishing a connection, evaluating performance metrics such as transmit power, spectrum characteristics, and error rate to ensure communication quality.

Set Up Test Environment

The RF non-signaling test firmware environment mainly includes a PC, tester, a USB-to-UART board, a device under test (DUT), and a shield box.

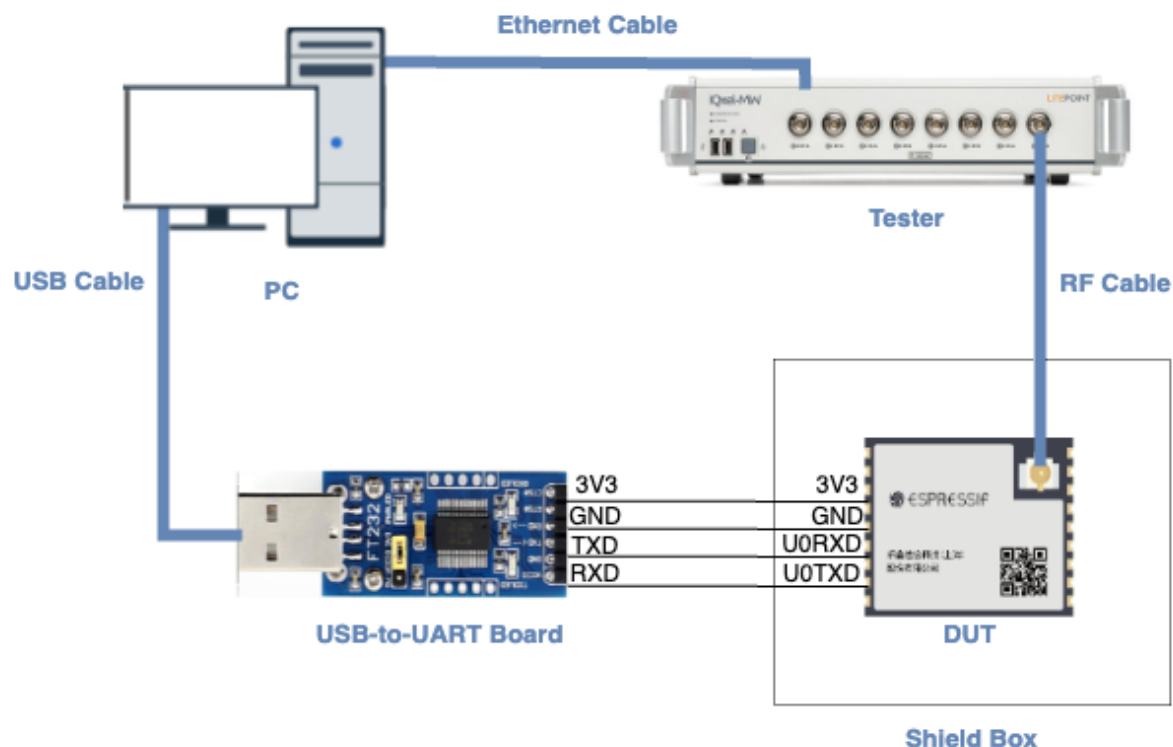


Fig. 34: Test Environment Setup

- **PC** is connected to the USB-to-UART board via USB and to the tester via an Ethernet cable. The PC needs to have the EspRFTTestTool toolkit, tester control software, and the driver for the USB-to-UART board installed.
- **Tester** is used to test the RF performance of the DUT in different modes. Typically, it is the WT-328/IQXel tester.
- **USB-to-UART board** is used to communicate between the PC and the DUT.
- **Device under test (DUT)** refers to a product designed based on the ESP32-C61 chip or module. It is connected to the USB-to-UART board via UART and to the tester via an RF connection cable. The DUT is usually placed inside a shield box.
- **Shield Box** is used to isolate external RF interference and ensure the stability of the test environment.

Note:

- The CHIP_EN pin of the DUT is pulled up by default. If it is not pulled up in the product design, you need to manually connect the CHIP_EN to the 3V3 pin.
- Some serial communication boards have already swapped RXD and TXD internally, so there is no need to reverse the connection. Adjust the wiring according to the actual situation.
- ESP32-C61 has a power-on self-calibration feature. The RF connection cable must be connected to the tester before the DUT is powered on for testing.

Conduction Test

- For modules without an onboard PCB antenna, the RF connection cable can be directly soldered to the antenna feed point of the module (as shown in the schematic diagram above).
- For modules with an onboard PCB antenna, cut the trace that connects to the PCB antenna feed point and solder the RF connection cable. The RF cable's shielding metal layer must be thoroughly soldered before connecting to the module's GND. The GND soldering point can be either the shield cover or the exposed GND layer on the PCB (after removing the green solder mask). Besides, it should be as close to the feed point as possible.



Fig. 35: Soldering RF Connection Cable to Module with Onboard PCB Antenna

Flash Firmware

1. Open [EspRFTestTool](#).
2. Set ChipType, COM, BaudRate, and click Open to open the COM port.

Note: Set BaudRate to 115200

3. Flash Contact Espressif for the latest firmware to Flash via UART.

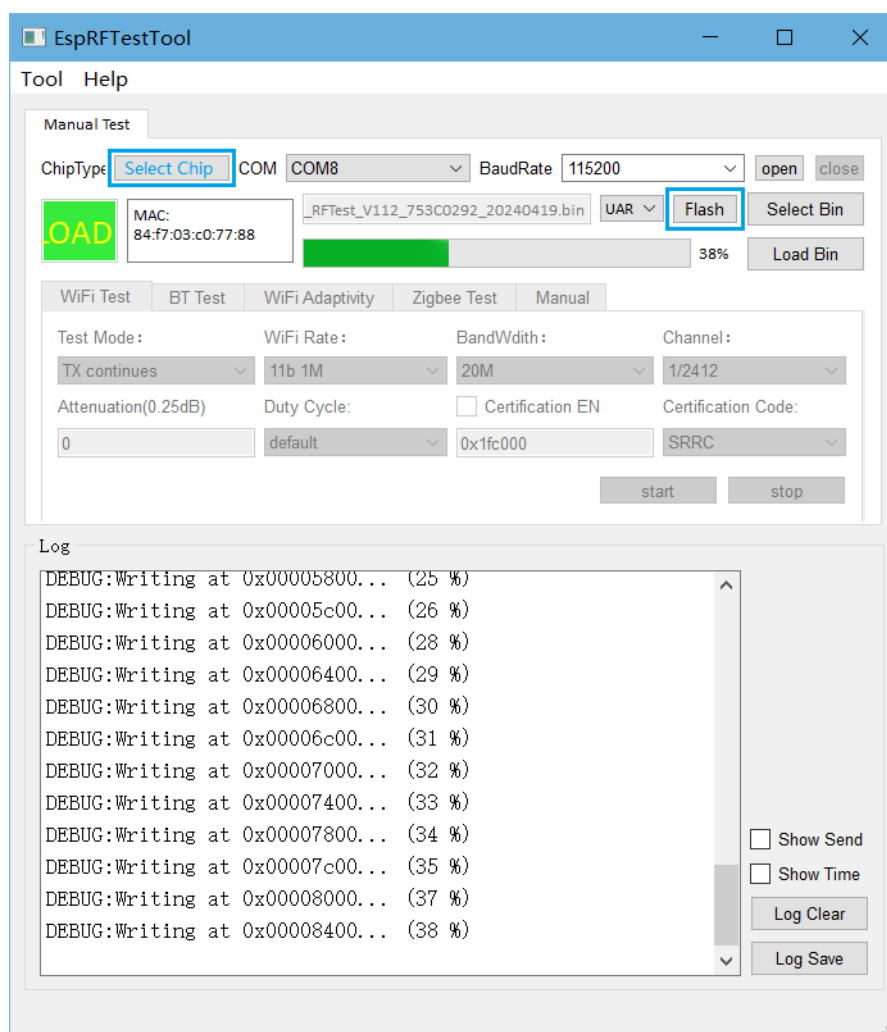


Fig. 36: ESPRFTTestTool Configuration

4. After the firmware flashing is completed, pull the boot pin high or leave it floating. The chip will enter the working mode after power-off restart.

Note: If you use the flash download tool to flash the firmware, change the flash address of ESP32-C61 to 0x0.

Start Testing

Bluetooth LE TX Performance Test

- **Test Mode:**
 - BLE50 TX: Used for TX performance tests;
 - BLE50 TX continue: High packet duty cycle used for certification testing.
- **Power Level:** Set the Bluetooth LE TX power level, supporting 0~15 level test
- **Channel:** Set the Bluetooth LE test channel
- **Hoppe:** Enable hopping function. Default: disabled
- **Ulap:** Set the Bluetooth address. Default: disabled
- **Itaddr:** Set the logical TX address. Default: disabled
- **Syncw:** Set the identity code of the packet file. Default: syncw=0x71764129
- **Payload length:** Set the payload length. Default: 250
- **Data Rate:** Set the packet TX rate and encoding sequence. It supports four rates, including BLE 1M, 2M, 125K, and 500K. It supports three encoding sequences, including 1010, 11110000, prbs9.

After clicking `start`, the Bluetooth LE TX parameter description is displayed in the log window, similar to the following:

```
fcc_le_tx_syncw:txpwr=15,chan=0,len=250,data_type=0,syncw=0x71764129,rate=0,tx_
↳num=0,contin_en=0,delay=0,hopp_en=0
```

This indicates that the Bluetooth LE packet TX is normal, and at this point, the tester can be used to test the TX performance.

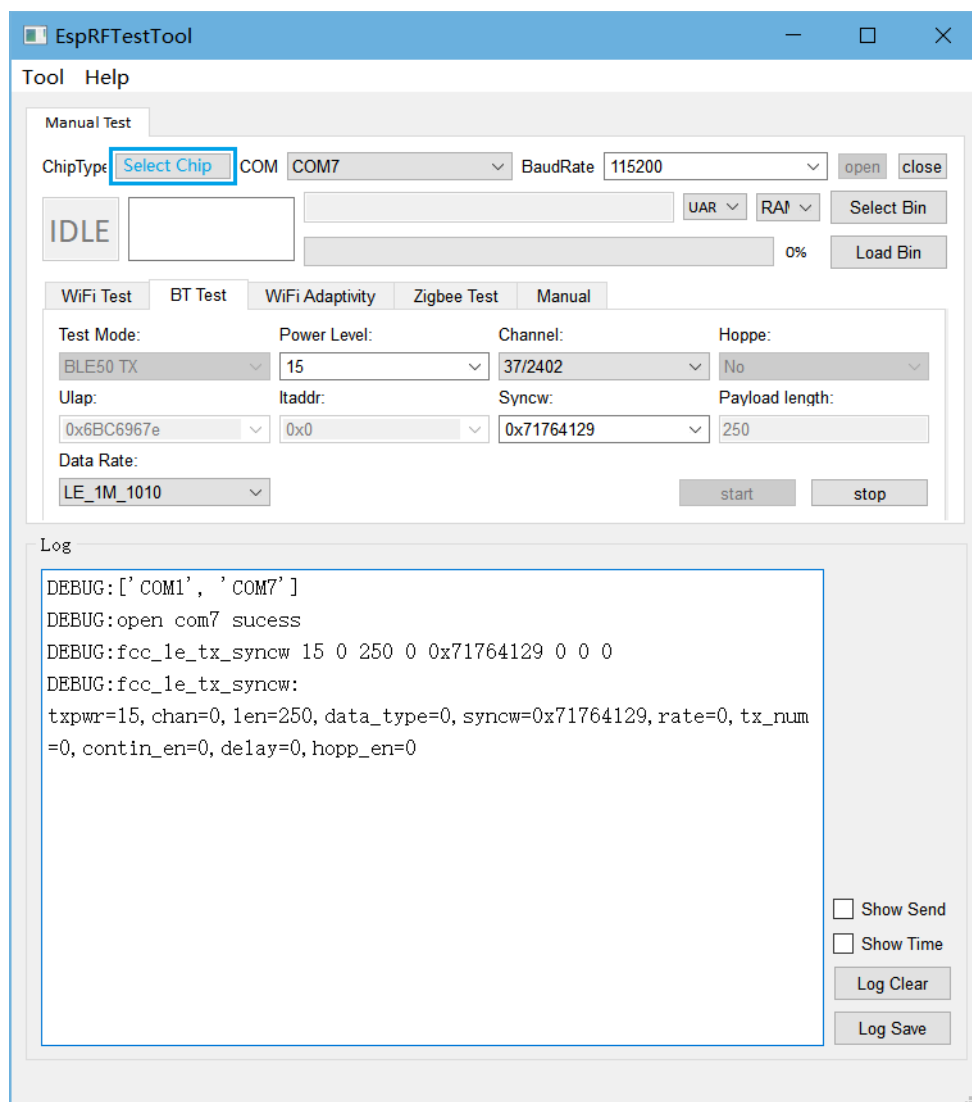


Fig. 37: Bluetooth LE TX Performance Test

Bluetooth LE RX Performance Test

- **Test Mode:** Select BLE50 RX for Bluetooth LE RX performance test.
- **Channel:** Set the Bluetooth LE test channel.
- **Syncw:** Set the identity code of the package file. Default :syncw=0x71764129.
- **Data Rate:** Set the package RX rate. The default encoding sequence is prbs9.

After clicking `start`, use the tester to send packages on the test channel. Click `stop` after completion. The package RX information is displayed in the log window, similar to the following:

```
3e8 3e8 0 0 0 0 0 0 0 0 p -53276 -24131 0 29422
```

Among them:

- The 1st parameter Res[0] (hexadecimal) represents the total number of packages received in this test. In this test, the total number of packages is 3e8.
- The 2nd parameter Res[1] (hexadecimal) represents the number of packages received at the corresponding rate in this test. In this test, the number of packages at the corresponding rate is 3e8.
- The 12th parameter Res[11] (in decimal) represents the RSSI of this test received correct package. In this test, the RSSI is -53276.

Based on the above parameters, you can calculate:

- Packet loss rate PER = $[1 - (\text{Res}[1] / \text{Sent_Packet_Numbers})] * 100\% \leq 30.8\%$
- RSSI of each package = $\text{Res}[11] / (\text{Res}[1])$

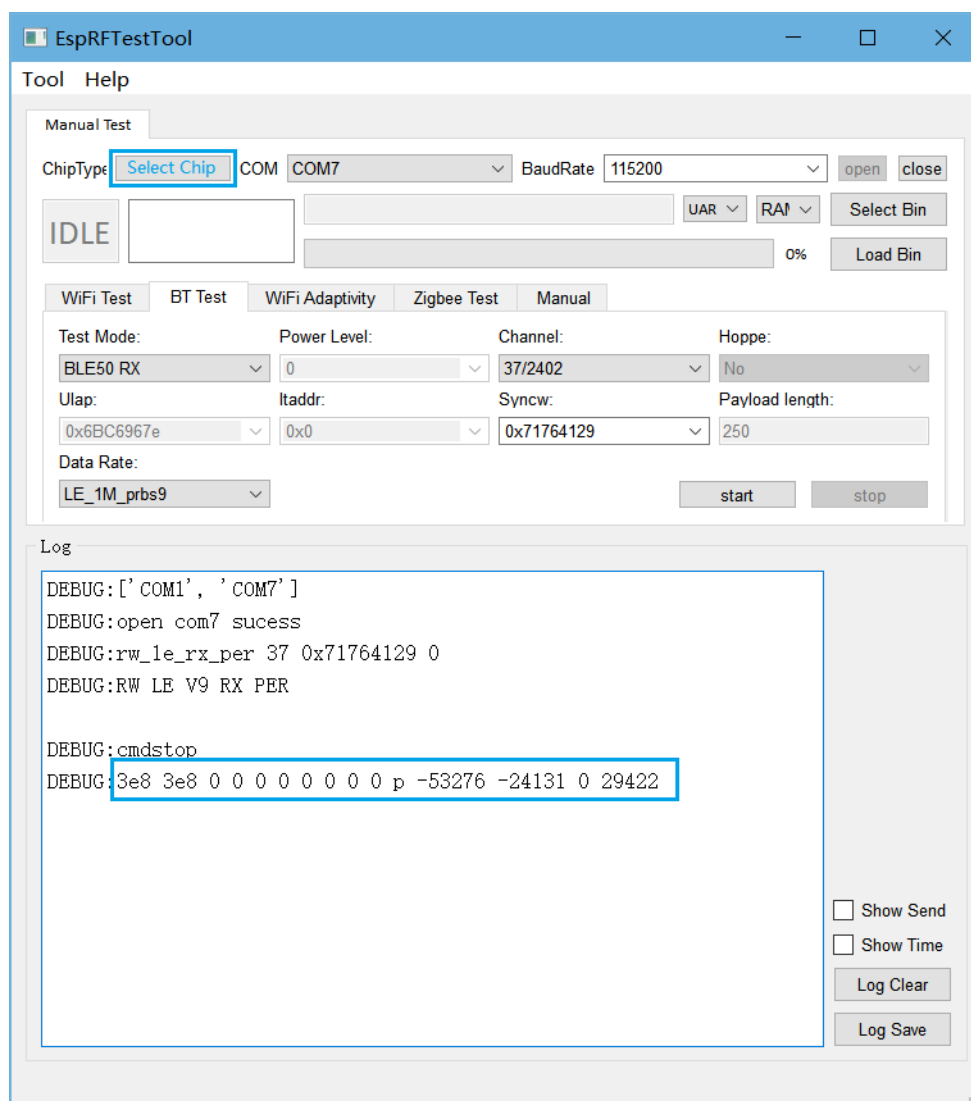


Fig. 38: Bluetooth LE RX Performance Test

Appendix

This appendix provides reference for RF debugging or testing.

Bluetooth LE TX Power Level This table shows the mapping between power levels and their target power for Bluetooth LE on ESP32-C61, serving as a reference for RF debugging or testing.

Table 3: ESP32-C61 Bluetooth LE TX Power Level

Power Level	Bluetooth LE TX Power (dBm)
0	-15
1	-15
2	-15
3	-15
4	-12
5	-9
6	-6
7	-3
8	0
9	3
10	6
11	9
12	12
13	15
14	18
15	20

Bluetooth LE 5.0 PHY Channel and Index For Bluetooth LE, the EspRFTTestTool toolkit uses the channel index to identify channels.

Table 4: ESP32-C61 Bluetooth LE 5.0 PHY Channel and Index

PHY Channel	RF Center Frequency (MHz)	Channel Index
0	2402	37
1	2404	0
2	2406	1
...	...	...
11	2424	10
12	2426	38
13	2428	11
14	2430	12
...	...	...
38	2478	36
39	2480	39

3.6 Bluetooth LE DTM Test

The Bluetooth LE DTM Test evaluates the RF performance of devices by directly controlling the device to enter specific transmit or receive modes, accessing key metrics like transmit power, reception sensitivity, and spectrum characteristics.

Set Up Test Environment

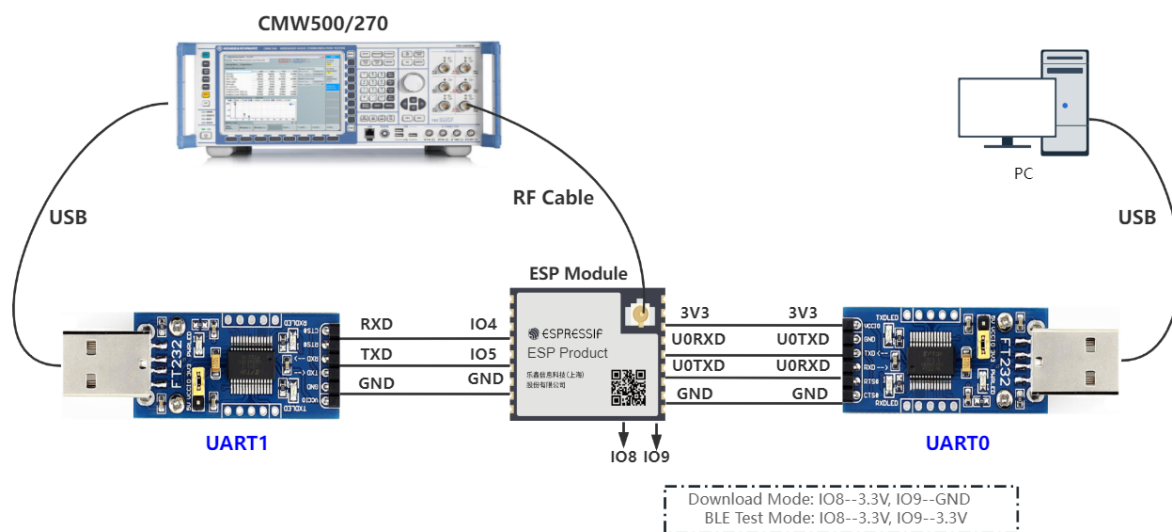


Fig. 39: Test Environment Setup

- **PC** is connected to the USB-to-UART board via USB. The PC needs to have the EspRFTTestTool toolkit, tester control software, and the driver for the USB-to-UART board installed.
- **Tester** is used to test the RF performance of the device under test (DUT) in different modes. It connects to DUT via an RF connection cable to transmit RF signals. Typically, it is CMW500, CMW270, or Bluetooth tester CBT.
- **USB-to-UART board** is used to communicate between the computer and the DUT, as well as between the tester and the DUT.
- **Device under test (DUT)** refers to a product designed based on the ESP32-C61 chip or module.

Note:

- The CHIP_EN pin of the DUT is pulled up by default. If it is not pulled up in the product design, you need to manually connect the CHIP_EN to the 3V3 pin.
- Some serial communication boards have already swapped RXD and TXD internally, so there is no need to reverse the connection. Adjust the wiring according to the actual situation.
- ESP32-C61 has a power-on self-calibration feature. The RF connection cable must be connected to the tester before the DUT is powered on for testing.

Conduction Test

- For modules without an onboard PCB antenna, the RF connection cable can be directly soldered to the antenna feed point of the module (as shown in the schematic diagram above).
- For modules with an onboard PCB antenna, cut the trace that connects to the PCB antenna feed point and solder the RF connection cable. The RF cable's shielding metal layer must be thoroughly soldered before connecting to the module's GND. The GND soldering point can be either the shield cover or the exposed GND layer on the PCB (after removing the green solder mask). Besides, it should be as close to the feed point as possible.

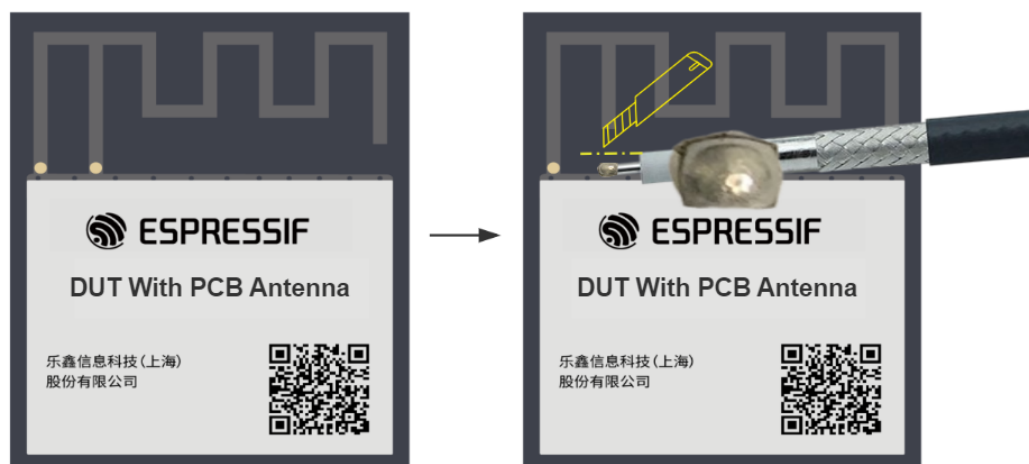


Fig. 40: Soldering RF Connection Cable to Module with Onboard PCB Antenna

Flash Firmware

1. Open [DownloadTool](#).
2. Set ChipType, Com Port, Baud Rate, click Open, and select to download to Flash.
3. Flash the Contact Espressif for the latest firmware bin file to the following address via UART.

Bin File	Flash Address
Contact Espressif for the latest firmware	0x0

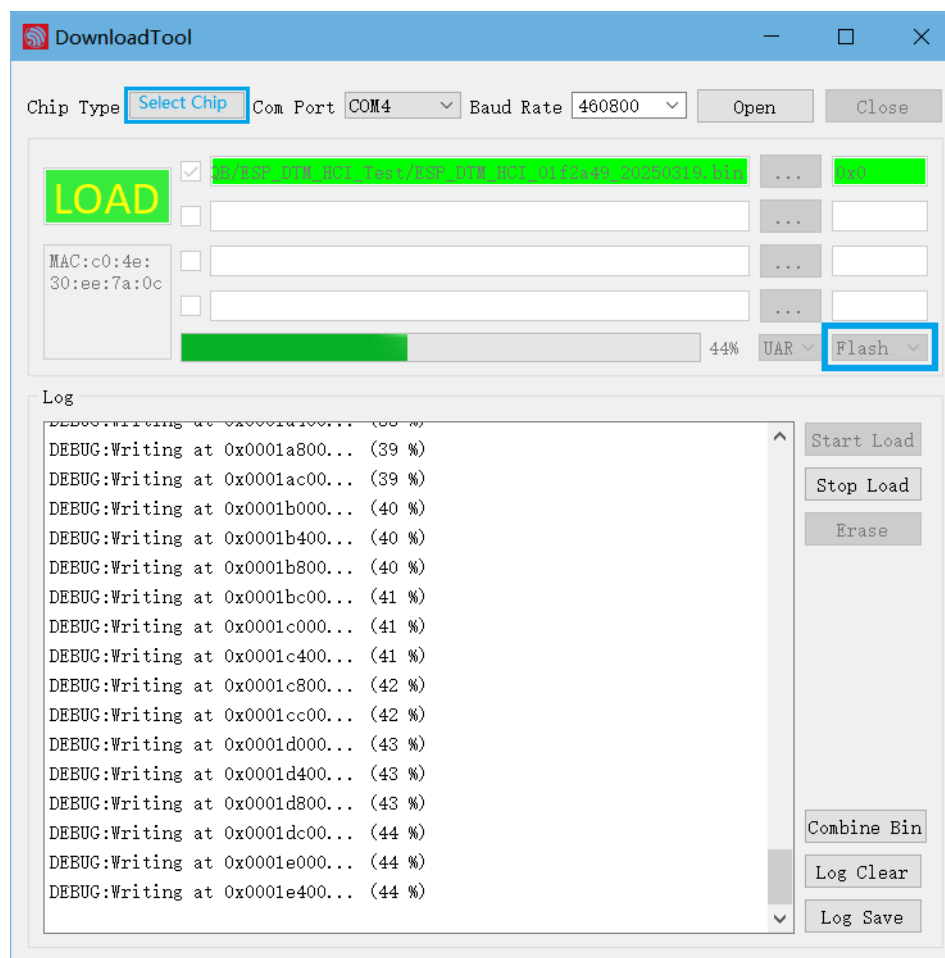


Fig. 41: Flash Firmware

After the flash process is completed, pull up or leave the boot pin unconnected. After the chip restarts and enters the working mode, continue with the following steps for testing.

Start Testing

The connection methods between the DUT and the tester includes HCI and 2-wire, with HCI being the default option.

Based on the hardware connections described above, you can verify whether the firmware flashing was successful by checking the output from the UART0 serial port.

Upon powering on, the device defaults to a power level of 12 dBm, operates without flow control, and uses a baud rate of 115200 for initialization. No commands are required, so you can directly begin the DTM test.

To adjust the relevant settings of UART1, you can input the corresponding commands in real time through the UART0 port:

```
// Configure TX output power. The supported power adjustment range is from 0 to 15_
↪levels.
set_ble_tx_power -i 15

// Get the current configuration power of BLE.
get_ble_tx_power

// Configure UART1, set TX pin to GPIO4 and RX pin to GPIO5.
reconfig_dtm_uart_pin -t 4 -r 5
```

Appendix

This appendix provides the mapping of power levels and target power of ESP32-C61 for RF debugging or testing reference.

Bluetooth LE Transmit Power Level This list provides the mapping of power levels and target power of ESP32-C61 Bluetooth LE for RF debugging or testing.

Table 5: ESP32-C61 Bluetooth LE Transmit Power Levels

Power Level	Bluetooth LE Transmit Power (dBm)
0	-15
1	-15
2	-15
3	-15
4	-12
5	-9
6	-6
7	-3
8	0
9	3
10	6
11	9
12	12
13	15
14	18
15	20

3.7 Bluetooth LE Adaptivity Test

The Bluetooth LE Adaptivity Test ensures that the device operates using frequency hopping and the Power Spectral Density (PSD) of the Bluetooth LE signal exceeds 10 dBm/MHz, meeting specific parameter requirements to prevent interference with other wireless devices.

Note:

- If the PSD of the Bluetooth LE signal transmitted by the device is less than 10 dBm/MHz, the interference mitigation techniques with the equivalent occupancy rate of no more than 10% can be applied. In this case, the Bluetooth LE Adaptivity Test is not required.
 - If the PSD of the Bluetooth LE signal transmitted by the device exceeds 10 dBm/MHz, the Listen Before Talk (LBT) mechanism based on frequency hopping can be used to perform the Bluetooth LE Adaptivity Test.
-

Set Up Test Environment

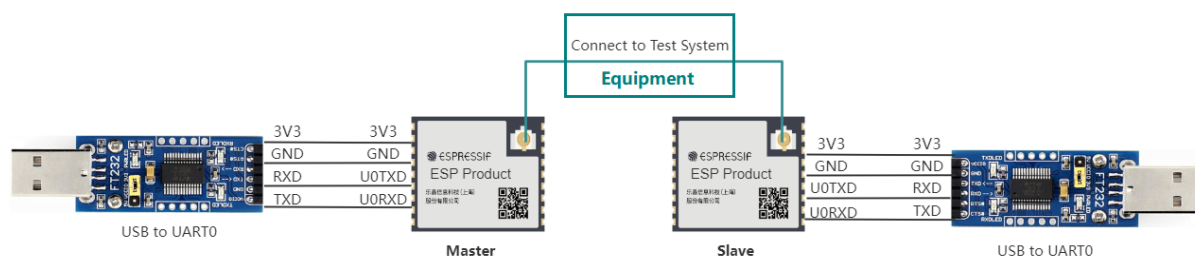


Fig. 42: Test Environment Setup

- In the test, the ESP32-C61 module is used as the test device (Slave) to establish a connection with the device under test (Master). Both the Slave and Master are flashed with the same firmware, but they can be distinguished using the serial port commands.
- The **Test System** refers to the system that performs adaptivity test. Once the Master and Slave are successfully connected via the serial port command, the test can begin.

Note:

- The CHIP_EN pin of the device under test is pulled up by default. If it is not pulled up in the product design, you need to manually connect the CHIP_EN to the 3V3 pin.
- Some serial communication boards have already swapped RXD and TXD internally, so there is no need to reverse the connection. Adjust the wiring according to the actual situation.
- ESP32-C61 has a power-on self-calibration function, so the RF connection cable must be connected to the tester before the device under test is powered on for testing.

Flash Firmware

1. Open [DownloadTool](#).
2. Set ChipType, Com Port, Baud Rate, click Open, and select to download to Flash.
3. Flash Contact Espressif for the latest firmware to 0x0 via UART.

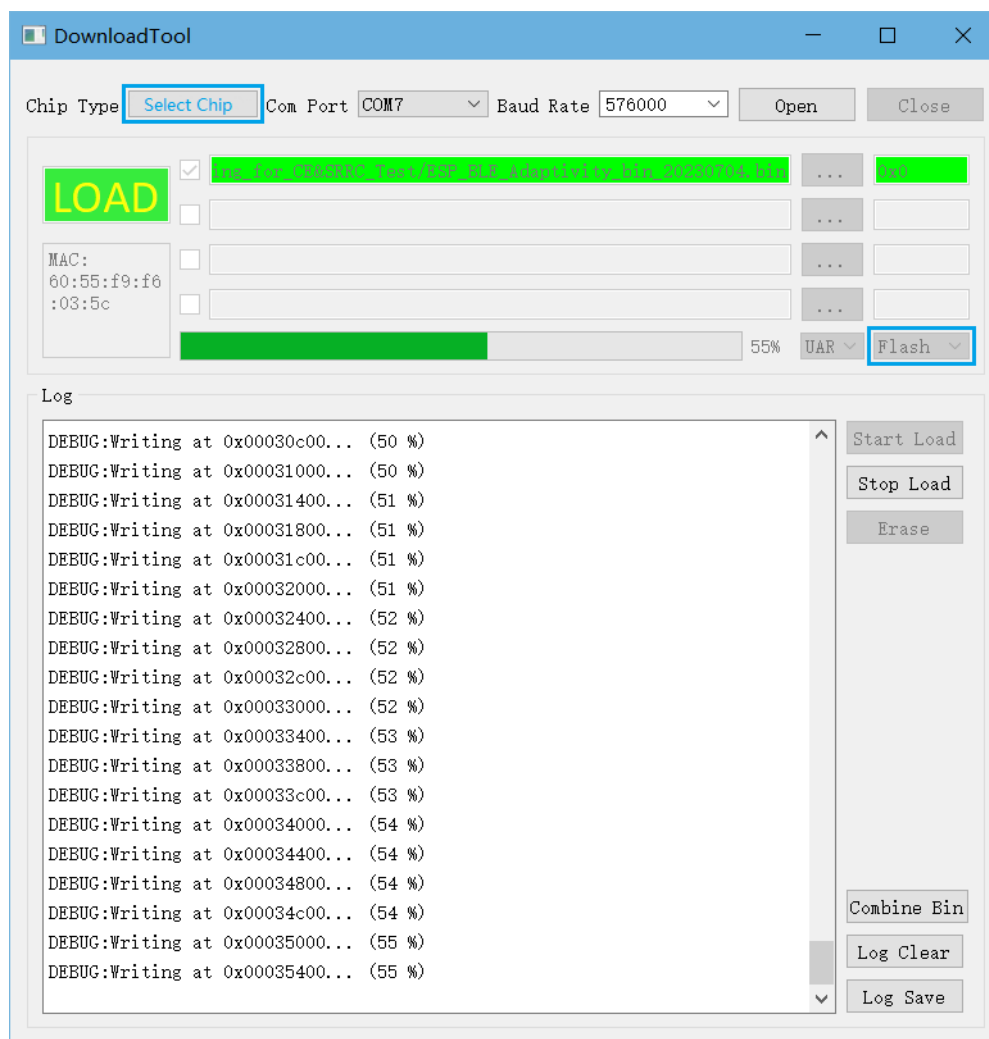


Fig. 43: Flash Firmware

After the flashing is completed, continue the following steps for testing.

Start Testing

Bluetooth LE Adaptivity Test requires inputting corresponding serial port commands in both Master and Slave devices to establish a connection for testing.

Open the serial port assistant and enter the corresponding commands on the Slave and Master devices in turn:

1. Slave Device

```
//Start advertising on the test device
bleadv -C -z start -t 19 -u 13
```

2. Master Device

```
//Establish a connection with a data rate of 1 Mbps (to configure 2 Mbps, change
parameters to "-x 2 -y 2"), set power level to 13
bleconn -T -z start -x 1 -y 1 -n 1 -i 0x6-0x6 -v 13

//Configure power, default is set to level 13 (the parameter after "-e" should be
consistent with the one after "-v" in the previous command)
blehci -S -z etxp -t 4 -h 1 -e 13
```

(continues on next page)

(continued from previous page)

```
//Set MTU
gattc -C -m 512 -p 0x10 -r c0:11:11:11:11:11 -b 1

//Send data
gattc -W -z char -p 0x10 -s 0xA002 -c 0xC317 -l 490 -n 0xFFFFFFFF -w 1 -r_
↪c0:11:11:11:11:11 -g 1 -b 1
```

3. Other Operation Commands

```
//Disconnect
bleconn -D -z all

//Reboot the module
reboot
```

After entering the above commands, you can continue with the Bluetooth LE Adaptivity Test.

3.8 Bluetooth LE Blocking Test

The Bluetooth LE Blocking Test evaluates the stability and performance of a device in environments with interference from other wireless signals. It ensures the device meets relevant standards for interference resistance.

Test Methods for Bluetooth LE Blocking

The test can be performed in two modes:

1. Non-signaling mode

In this mode, a fixed frequency is selected to test the device's resistance to interference. For details, please refer to [Bluetooth LE Non-Signaling Test](#).

2. Direct Test Mode (DTM)

DTM mode allows low-level control of the device, enabling the introduction of interference signals to access its interference resistance. For details, please refer to [Bluetooth LE DTM Test](#).

4 RF Test Certification

4.1 CE Certification

CE Certification (Conformité Européene Mark) is a mandatory certification by the EU, confirming compliance with safety, health, and environmental protection standards.

The CE certification of RF products requires non-signaling, adaptivity, and blocking tests:

- [Wi-Fi Non-Signaling Test](#)
- [Wi-Fi Adaptivity Test](#)
- [Wi-Fi Blocking Test](#)
- [Bluetooth LE Non-Signaling Test](#)
- [Bluetooth LE DTM Test](#)

- [Bluetooth LE Blocking Test](#)
- [Bluetooth LE Adaptivity Test](#)

4.2 FCC Certification

FCC Certification (Federal Communications Commission Certification) is a mandatory certification by the U.S. Federal Communications Commission, ensuring compliance with regulations on radio spectrum use, electromagnetic compatibility, and RF radiation.

The FCC certification of RF products requires passing relevant non-signaling tests:

- [Wi-Fi Non-Signaling Test](#)
- [Bluetooth LE Non-Signaling Test](#)

4.3 SRRC Certification

The SRRC (State Radio Regulatory Commission) Certification is a mandatory certification for radio equipment in China, ensuring compliance with national radio management regulations to avoid electromagnetic interference.

The SRRC certification of RF products requires related non-signaling and adaptivity tests:

- [Wi-Fi Non-Signaling Test](#)
- [Wi-Fi Adaptivity Test](#)
- [Bluetooth LE Non-Signaling Test](#)

Warning: This document is not updated for ESP32C61 yet, so some of the content may not be correct.
This warning was automatically inserted due to the source file being in the `add_warnings_pages` list.

5 WFA Certification and Testing Guide

5.1 Overview

This section provides guidance on obtaining Wi-Fi Alliance (WFA) certification for products based on Espressif chips. It focuses on the QuickTrack process to help you efficiently achieve WFA certification.

Required tools and firmware:

- [Flash Download Tool](#)
- `espsigma` tool and firmware

5.2 Introduction to WFA Certification

Certification Process

The WFA certification typically follows these steps:

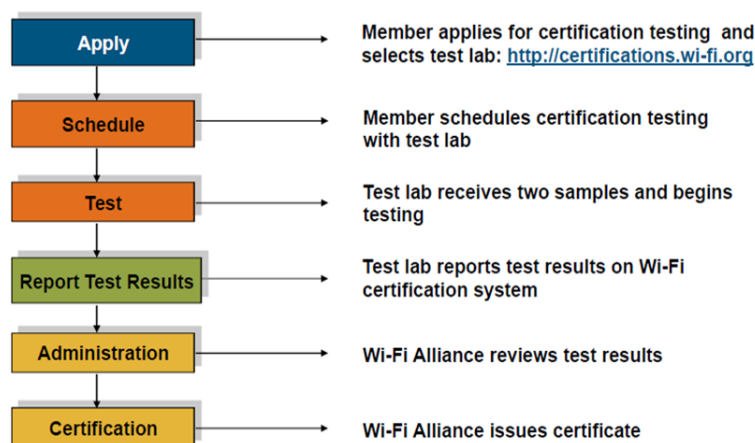


Fig. 44: Standard Process

1. Submit the certification application and choose an Authorized Test Laboratory (ATL), which receives the Certification ID (CID).
2. Send the device to ATL for testing.
3. ATL conducts the necessary test.
4. ATL provides the test results.
5. WFA issues the certification.

Certification Types

1. New Certification

Choose this option if the product has not been Wi-Fi certified before.

2. Additional Certification

Choose this option if the product is already certified but needs to test new features.

3. Re-Certification

If there are changes to the firmware or software that affect Wi-Fi functionality, re-certification is needed. This includes:

- Small hardware changes or updates to device software (e.g., operating system or drivers)
- Firmware changes or minor software modifications that affect Wi-Fi operation (even small updates or bug fixes)
- Changes that don't affect Wi-Fi functionality must be reviewed by ATL to determine if testing is required

4. Derivative Certification

This applies to derivative products based on a source certification. The derivative product must be functionally consistent with the source product, and technical details must be provided to verify eligibility.

Note: WFA certification primarily targets products operating in Wi-Fi 802.11a/b/g/n modes, typically using 2.4 GHz or 5 GHz radio frequency bands. This includes devices like wireless routers, smartphones, home appliances, computers, network infrastructure, and consumer electronics.

5.3 Espressif Product Certification Process

Certification Method

- **Espressif Modules:** Generally, these follow the **New Certification** path.
- **Products based on Espressif Chips:** **QuickTrack** path is recommended.

The relationship between the two is as follows:



Fig. 45: New Certification and QuickTrack

Once an Espressif module completes a new certification, Espressif stores the test data and generates a **Qualified Solution**. You can leverage this solution to streamline your certification process.

New Certification

The test items for Espressif modules are shown in the figure below.

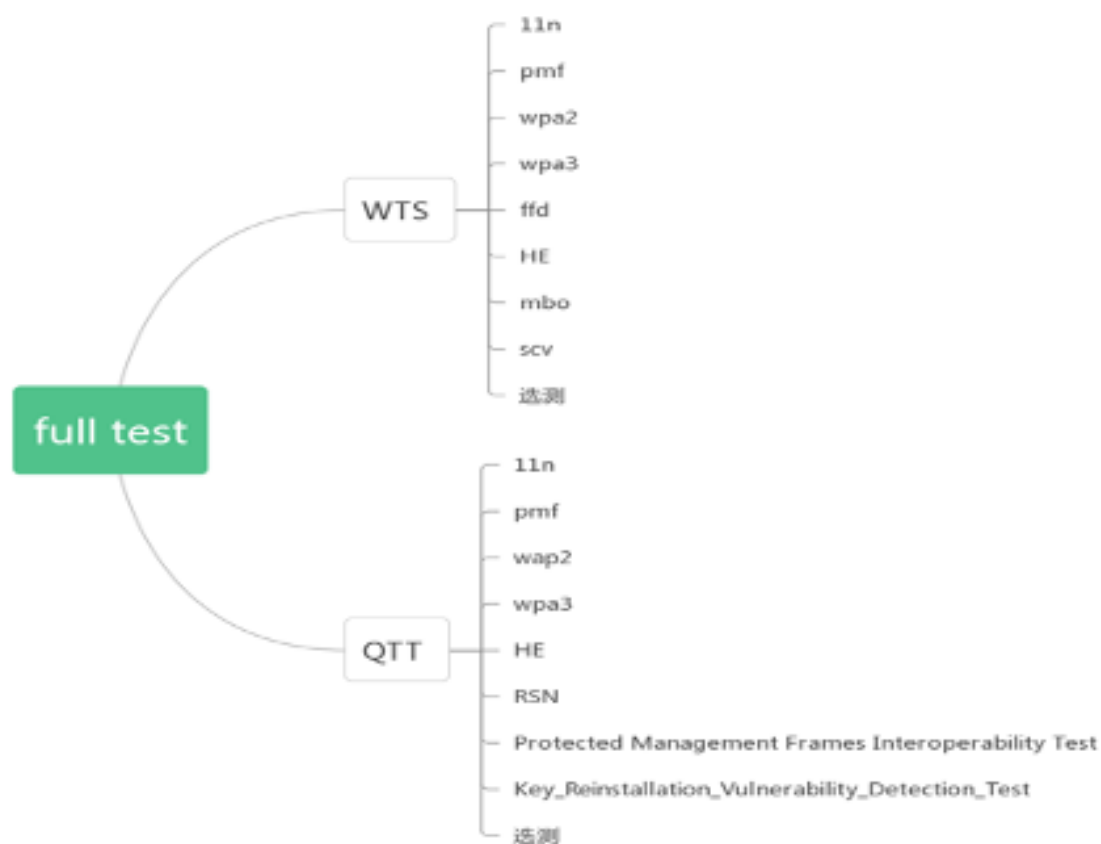


Fig. 46: Full-test Test Items

The WFA testing includes two parts: **WTS (Sigma Tool Test Items)** and **QTT (QuickTrack Test Items)**. While some test items are same, the test cases differ between the two.

QuickTrack

QuickTrack is a streamlined Wi-Fi certification method aimed at reducing testing and certification costs while speeding up the process. This method is designed for products built using a **Qualified Solution**.

To achieve QuickTrack:

1. Select components or solutions from a list of **Qualified Solutions** that meet your product requirements.
2. Perform consistency tests to ensure the components or solutions meet the **Qualified Solution** criteria.
3. Complete testing using tools provided by WFA, either in-house or through ATL.
4. Submit the test results for review by WFA.
5. Once the test results are approved by WFA, the product will receive Wi-Fi certification.

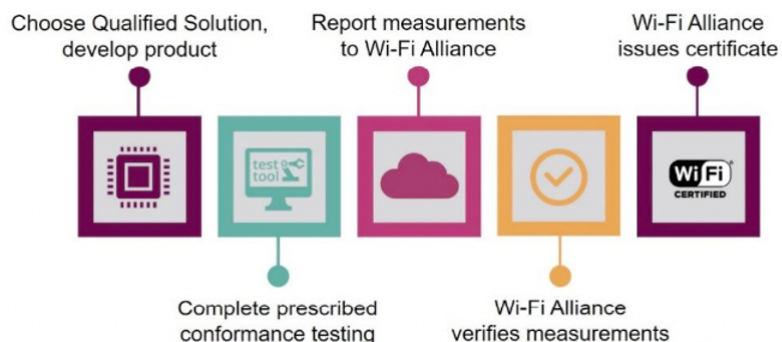


Fig. 47: Quick Certification Process

Advantages of QuickTrack QuickTrack reduces costs and testing time, helping you achieve WFA certification faster. For example, the ESP32-C2 module's full certification test takes about 7.5 days.

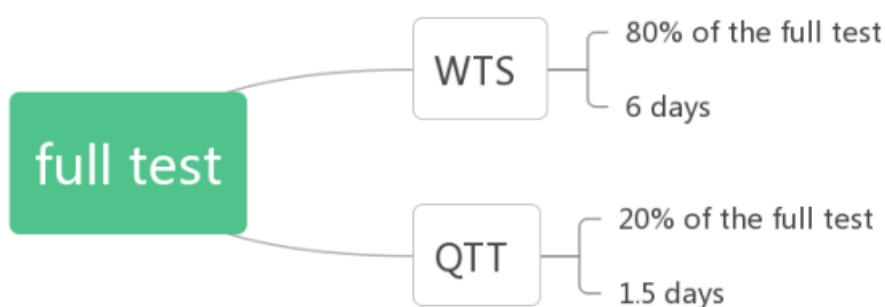


Fig. 48: Full-test Testing Time

If you choose QuickTrack, you must first confirm the product information below:

Wireless Chipset ESP32-C2	Wi-Fi Component Operating System Free RTOS
Wi-Fi Component Firmware Version A	Physical Interface UART
Driver V7.0	RF Components RF matching, RF switch connector
	Antenna PCB Antenna
RF Architecture	
Bands Supported	Transmit (Tx)
2.4 GHz	1
	Receive (Rx)
	1

Fig. 49: Product Information

- Using QuickTrack, you need only 1.5 days for QTT testing if your product differs from ESP32-C2.
- If no changes are made, certification can be obtained without further testing by simply paying the certification fee.

The comparison between QuickTrack and the ordinary certification method is as follows:

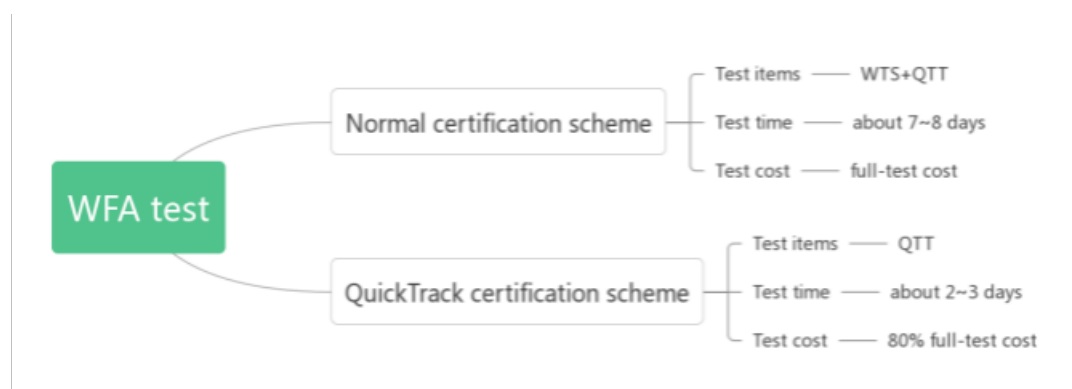


Fig. 50: Comparison Between Ordinary Certification and QuickTrack

Note: The test items mentioned refer only to the testing portion of the certification process. The full WFA certification process, including submission and approval, can take up to 40 days for standard certification. QuickTrack reduces this to approximately 10 days, saving around 70% of the time.

Current QuickTrack Status for Espressif Chips

Currently, both ESP32-C2 and ESP32-C6 have completed QuickTrack **Qualified Solution** certification.

5.4 WFA Testing

1. Submit CID Information

You can fill in CID information according to requirements by referring to [Wi-Fi Alliance CID Filling Guide](#) and [Espressif Module Filling Method](#).

2. Flash Firmware

Flashing on Windows

- Open the `flash_download_tool_3.9.2.exe` application.
- Set `chipType` to the corresponding chip name and `workMode` to `develop`, then click OK.
- Choose the firmware and specify the flashing address. Select the port number, set baud to 115200, and click START to begin flashing.
- Once flashing is complete, `finish` will be displayed.

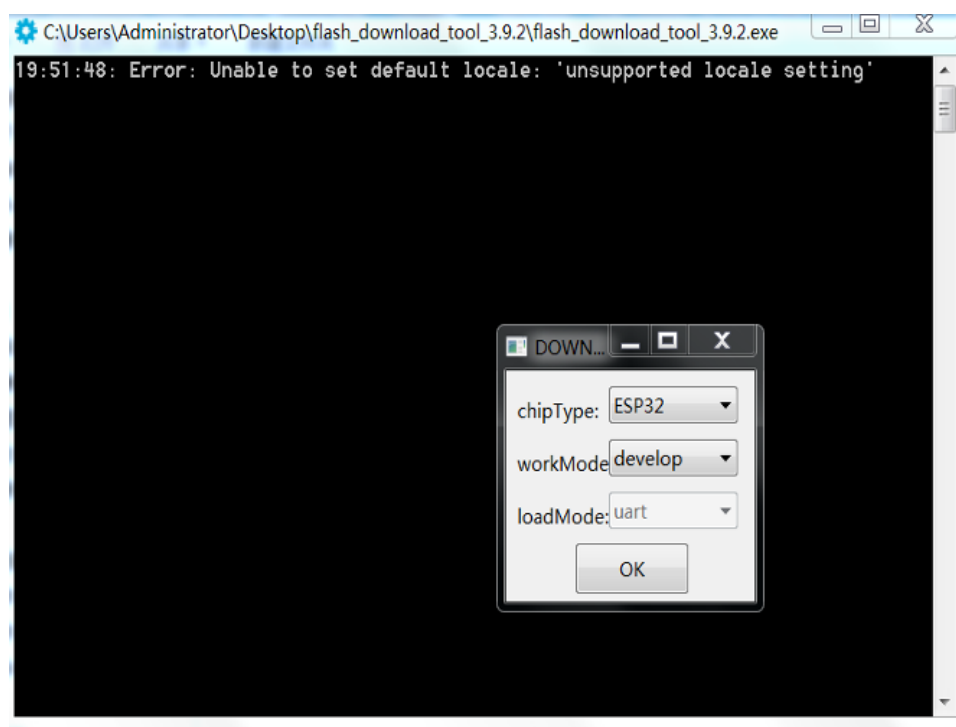


Fig. 51: Flash Configuration

Flash the following firmware to the corresponding address:

- `bootloader.bin` 0x0
- `espsigma.bin` 0x10000
- `partition.bin` 0x8000

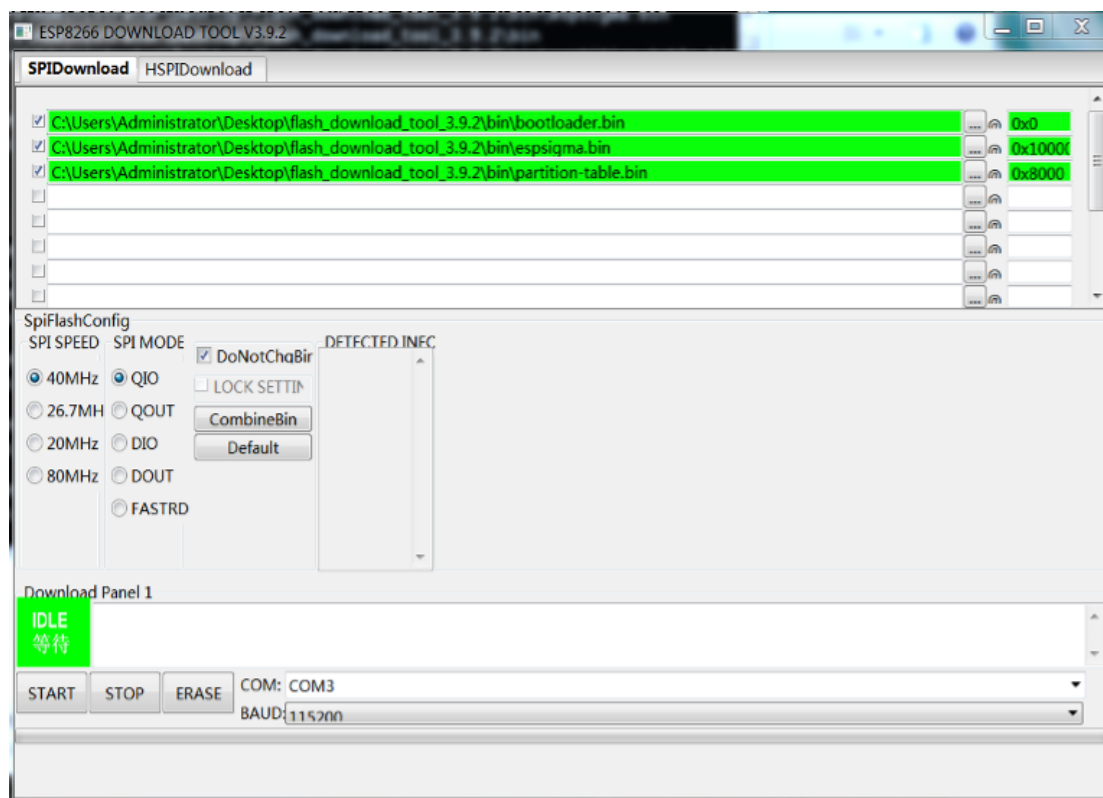


Fig. 52: Flash Firmware

Flashing on Ubuntu

- Install Python 3.7

```
cd espsigma_qt/espsigma
./tools/setup/setup_pyenv_python.sh
source ~/.pyenv/activate
```

- Install the flash tool

```
pip install esptool
```

- Start flashing

```
esptool.py -p /dev/ttyUSB0 --chip=auto write_flash 0x0 bootloader.bin 0x8000_
partition-table.bin 0x10000 espsigma.bin
```

3. Set Up the Test Environment

- Use Ubuntu 16.04 or higher
- Install Python 3.7

```
cd espsigma_qt/espsigma
./tools/setup/setup_pyenv_python.sh
source ~/.pyenv/activate
```

After installation, you can verify the Python version with `python -v`.

Note: When flashing firmware on a computer with an Ubuntu OS, the Python environment is already installed in the step of flashing firmware, so no further installation is needed. Only computers running Windows require this step

to install the Python environment.

Note: The Python version must be 3.7 or higher. If the version displayed in the terminal is incorrect, please the above command.

4. Start Testing

Test the WTS Part

- Open a terminal and navigate to the Sigma test tool directory

```
cd /espsigma_qt/espsigma/esp_sigma_ca
```

- Start the test

```
python espsigma.py --dut /dev/ttyUSB*
```

Note: * refers to the serial port number.

```

→ esp_sigma_ca git:(merging_c6_ax_changes) X python esp_sigma.py --dut /dev/ttyUSB1 -v
[scmd] version
[console] version
IDF Version:v5.2-dev-151-g9cfc9757b9-dirty
Chip info:
  model:Unknown
  cores:1
  feature:/802.11bgn/BLE/External-Flash:2 MB
  revision number:0
esp_sigma>
esp_sigma>
Got Response for version as version
IDF Version:v5.2-dev-151-g9cfc9757b9-dirty
Chip info:
  model:Unknown
  cores:1
  feature:/802.11bgn/BLE/External-Flash:2 MB
  revision number:0
esp_sigma>
esp_sigma>
[lcmd] git rev-parse --abbrev-ref HEAD
[resp] merging_c6_ax_changes

[lcmd] git rev-parse HEAD
[resp] 8e7cf533396db323e92cb835ba8055f3bd1d55bc

[scmd] settime 1685098860
[console] settime 1685098860
set time: 0
esp_sigma>
esp_sigma>
Got Response for settime 1685098860 as settime 1685098860
set time: 0
esp_sigma>
esp_sigma>

System Time set to : 2023-05-26 11:01:00.053476
[scmd] set_prog --program any
[console] set_prog --program any
I (4797) wifi:ifx:0, phymode(new:0x3, nvs:0x5)
cmd_wifi.c set_prog: Sta set BW to 40 Mhz
esp_sigma>
esp_sigma>
Got Response for set_prog --program any as set_prog --program any
I (4797) wifi:ifx:0, phymode(new:0x3, nvs:0x5)
cmd_wifi.c set_prog: Sta set BW to 40 Mhz
esp_sigma>
esp_sigma>
*****
Espressif Wi-Fi Alliance Sigma DUT Agent

WFA PROGRAM      : WPA3
DUT PORT         : /dev/ttyUSB0
IDF Version      : v5.2-dev-151-g9cfc9757b9-dirty
SIGMA CA         : merging_c6_ax_changes (8e7cf53)

```

Fig. 53: WTS Test

Test the QuickTrack Part

- Open a terminal and navigate to the Sigma test tool directory

```
cd /esp_sigma_qt/esp_sigma/esp_sigma_ca
```

- Start the test

```
python esp_sigma.py --quicktrack --dut/dev/ttyUSB *
```

Note: * refers to the serial port number.


```

test@FA000610: ~/espc6 x test@FA000610: ~/espc6 x test@FA000610: ~ x
→ esp_sigma_ca git:(merging_c6_ax_changes) X python3 espsigma.py --quicktrack --dut /dev/ttyUSB0 -v
[scmd] version
[console] version
IDF Version:v5.2-dev-151-g9cfc9757b9-dirty
Chip info:
  model:Unknown
  cores:1
  feature:/802.11bgn/BLE/External-Flash:2 MB
  revision number:0
espsigma>
espsigma>
Got Response for version as version
IDF Version:v5.2-dev-151-g9cfc9757b9-dirty
Chip info:
  model:Unknown
  cores:1
  feature:/802.11bgn/BLE/External-Flash:2 MB
  revision number:0
espsigma>
espsigma>
[lcmd] git rev-parse --abbrev-ref HEAD
[resp] merging_c6_ax_changes

[lcmd] git rev-parse HEAD
[resp] c097929210d94c2bc502293f052f98e8ffef827

[scmd] settime 1675189801
[console] settime 1675189801
set time: 0
espsigma>
espsigma>
Got Response for settime 1675189801 as settime 1675189801
set time: 0
espsigma>
espsigma>
System Time set to : 2023-01-31 18:30:01
Terminal prog --program any
[console] set_prog --program any
I (97658) wifi:ifx:0, phymode(new:0x3, nvs:0x5)
cmd_wifi.c set_prog: Sta set BW to 40 Mhz
espsigma>
espsigma>
Got Response for set_prog --program any as set_prog --program any
I (97658) wifi:ifx:0, phymode(new:0x3, nvs:0x5)
cmd_wifi.c set_prog: Sta set BW to 40 Mhz
espsigma>
espsigma>
*****
Espressif Wi-Fi Alliance QuickTrack Test Tool Sigma DUT Agent

WFA PROGRAM      : WPA3
DUT PORT         : /dev/ttyUSB0
IDF Version      : v5.2-dev-151-g9cfc9757b9-dirty
SIGMA CA        : merging_c6_ax_changes (c097929)

*****
>Info: ESP Sigma listening at /tmp/socket_test.s

```

Fig. 54: QuickTrack Test-1

- Open another terminal and navigate to the control app directory

```
cd /espsigma_qt/controlappc-2.0.0.9
```

- Start the control app

```
./app -p *
```

Note: * refers to the QuickTrack Test (QTT) port, e.g., 9005.

```

→ espigna_qt git:(merging_c6_ax_changes) X cd controlappc-2.0.0.9
→ controlappc-2.0.0.9 git:(merging_c6_ax_changes) X ./app -p 9004
Welcome to use QuickTrack Control App DUT version 2.1.0.42.

Use default interface parameters 2:wlan0,2:wlan1,5:wlan0,5:wlan1.

wlan0_bridge = br-wlan0.
Jun 06 15:59:34 controlappc. info QuickTrack control app running at: 9004
Jun 06 15:59:34 controlappc. info Wireless Interface:
Jun 06 15:59:34 controlappc. info interface_count=4
Jun 06 15:59:34 controlappc. info Interface Name: wlan0, Band: 2.4GHz, identifier -1
Jun 06 15:59:34 controlappc. info Interface Name: wlan1, Band: 2.4GHz, identifier -1
Jun 06 15:59:34 controlappc. info Interface Name: wlan0, Band: 5GHz, identifier -1
Jun 06 15:59:34 controlappc. info Interface Name: wlan1, Band: 5GHz, identifier -1
Jun 06 15:59:34 controlappc. info hostapd Path: /usr/local/bin/WFA-Hostapd-Supplicant/hostapd (hostapd)
Jun 06 15:59:34 controlappc. info wpa_supplicant Path: /usr/local/bin/WFA-Hostapd-Supplicant/wpa_supplicant (wpa_supplicant)
Jun 06 15:59:34 controlappc. info Hostapd Global Control Interface: /var/run/hostapd-global
Jun 06 15:59:34 controlappc. info Hostapd Control Interface: /var/run/hostapd/wlan0
Jun 06 15:59:34 controlappc. info WPA Supplicant Control Interface: /tmp/socket_test.s
^CJun 06 19:02:54 controlappc. info Signal 2 received - terminating

```

Fig. 55: QuickTrack Test-2

Please refer to the pictures for Quicktrack page settings

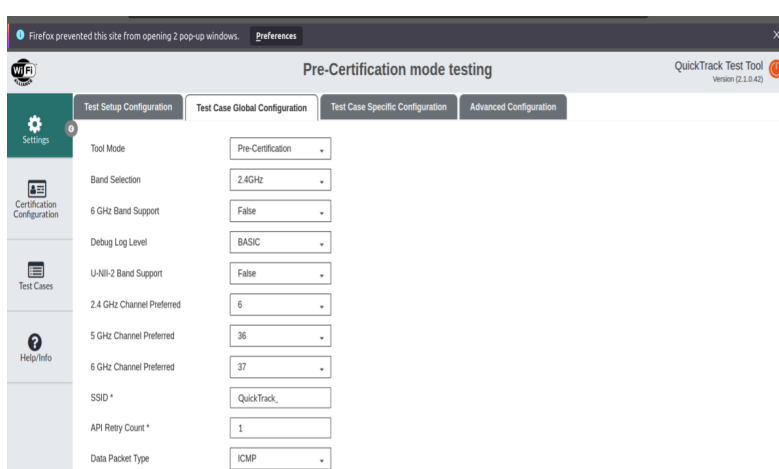


Fig. 56: QuickTrack Settings-1

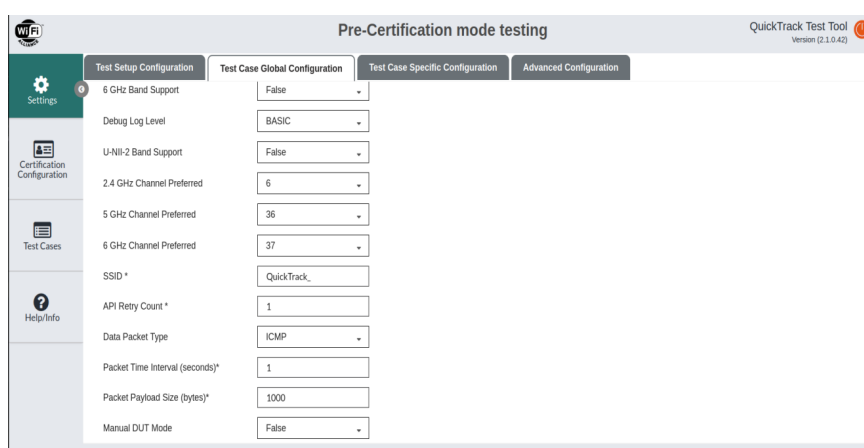


Fig. 57: QuickTrack Settings-2

The screenshot shows the 'Pre-Certification mode testing' interface of the QuickTrack Test Tool (Version 2.1.0.42). The left sidebar contains icons for Settings, Certification Configuration, Test Cases, and Help/Info. The main area is divided into four tabs: Test Setup Configuration, Test Case Global Configuration, Test Case Specific Configuration (selected), and Advanced Configuration. Under the 'Test Case Specific' tab, the 'Role' is set to 'Station'. The 'PHY Capabilities' dropdown is set to 'Wi-Fi 4'. The 'Security Capabilities' dropdown is set to 'WPA2-PMF', with 'WPA3-Personal' and 'RSN' also listed. A list of checkboxes on the right includes: 'Trigger unicast protected disconnect from GUI' (checked), 'PMK Caching support when disconnect triggered from STAUT' (checked), 'Support WEP default key 0 simultaneously with a pairwise key in a TSN network' (unchecked), 'Support Joint Multi-band RSNA' (unchecked), 'Support PeerKey handshake' (unchecked), 'Support signaling and payload protected (SPP) A-MSDU's transmission and reception' (unchecked), 'Only allow SPP A-MSDU's transmission and reception' (unchecked), 'Support protected block ack agreement capability (PBAC)' (unchecked), and 'Support Extended Key ID for individually addressed frames' (unchecked).

Fig. 58: QuickTrack Settings-3

The screenshot shows the 'Pre-Certification mode testing' interface of the QuickTrack Test Tool (Version 2.1.0.42). The left sidebar is the same as in Fig. 58. The main area is divided into four tabs: Test Setup Configuration, Test Case Global Configuration, Test Case Specific Configuration, and Advanced Configuration (selected). Under the 'Advanced Configuration' tab, there are several settings: 'Support protected WUR' (unchecked), 'Support secure LTF' (unchecked), 'Require MFP for preassociation ranging' (set to 'No'), 'Support protected block ack agreement capability (PBAC) as defined after 802.11-2020' (unchecked), 'Support signaling and payload protected (SPP) A-MSDU transmission and reception as defined after 802.11-2020' (unchecked), and 'WPA3 Supported DH Groups' (set to '19'). Below these, there is an 'Additional Capabilities' dropdown. Under the 'Enterprise' tab, the 'Authentication Server(AAA) IP Address' is set to '127.0.0.1' and the 'EAP-Method' is set to 'EAP_TLS'. There are also expandable sections for 'STA configurations' and 'Certificate path'.

Fig. 59: QuickTrack Settings-4

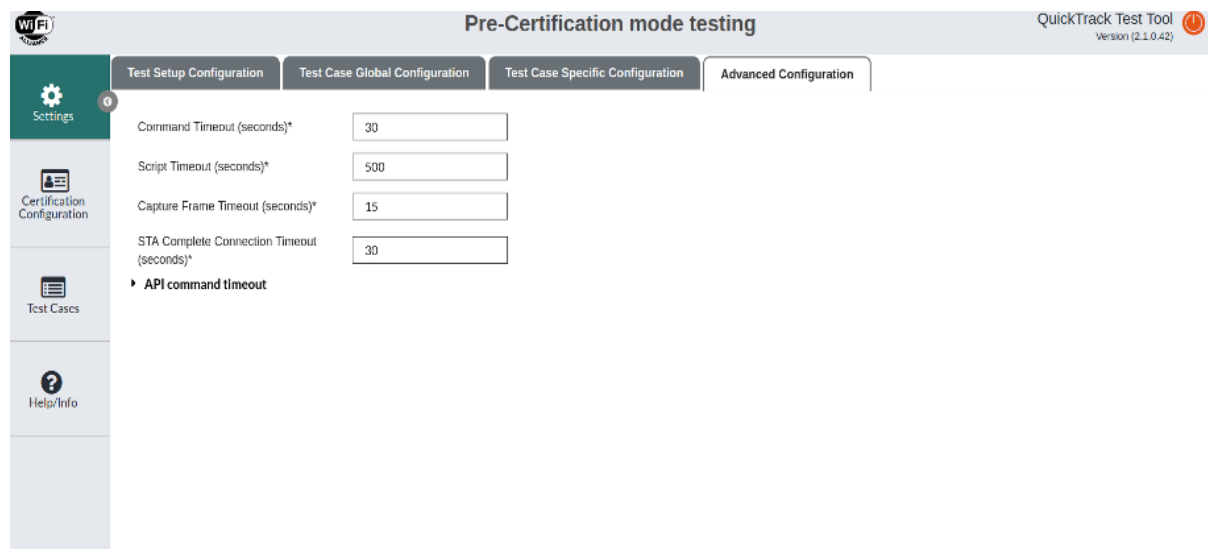


Fig. 60: QuickTrack Settings-5

6 Production Stage

For the production stage, this repository provides the following tools and resources designed to streamline the manufacturing process:

- [Flash Download Tool](#) is used to flash firmware onto flash. It supports multiple targets and configurations, enabling users to efficiently update firmware and debug devices.
- [Espressif Production Testing Guide](#) outlines the production testing schemes available for Espressif Wi-Fi products, thus providing reference for testing customer products during manufacturing.
- [Test Fixture Manufacturing Instruction](#) provides guidelines for manufacturing test fixtures used with Espressif's Wi-Fi modules. These standardized fixtures help prevent issues that may arise during production and testing.
- [Matter QR Code Generator](#) is used to generate Matter device networking QR codes, allowing users to quickly add devices to the smart home network by scanning the QR code, simplifying the device configuration and connection process.

Warning: This document is not updated for ESP32C61 yet, so some of the content may not be correct.

This warning was automatically inserted due to the source file being in the `add_warnings_pages` list.

7 Flash Download Tool User Guide

7.1 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

7.2 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:

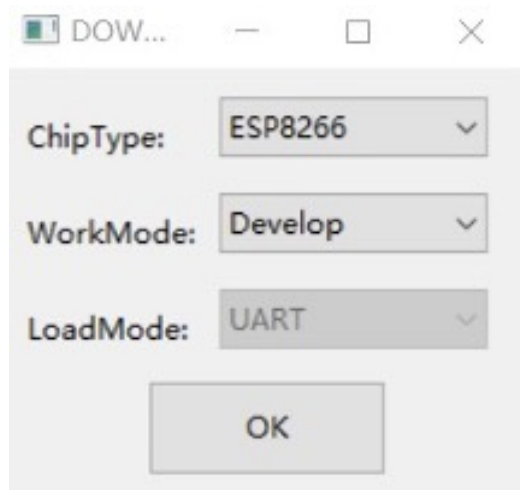


Fig. 61: Flash Download Tool Main Interface

- **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

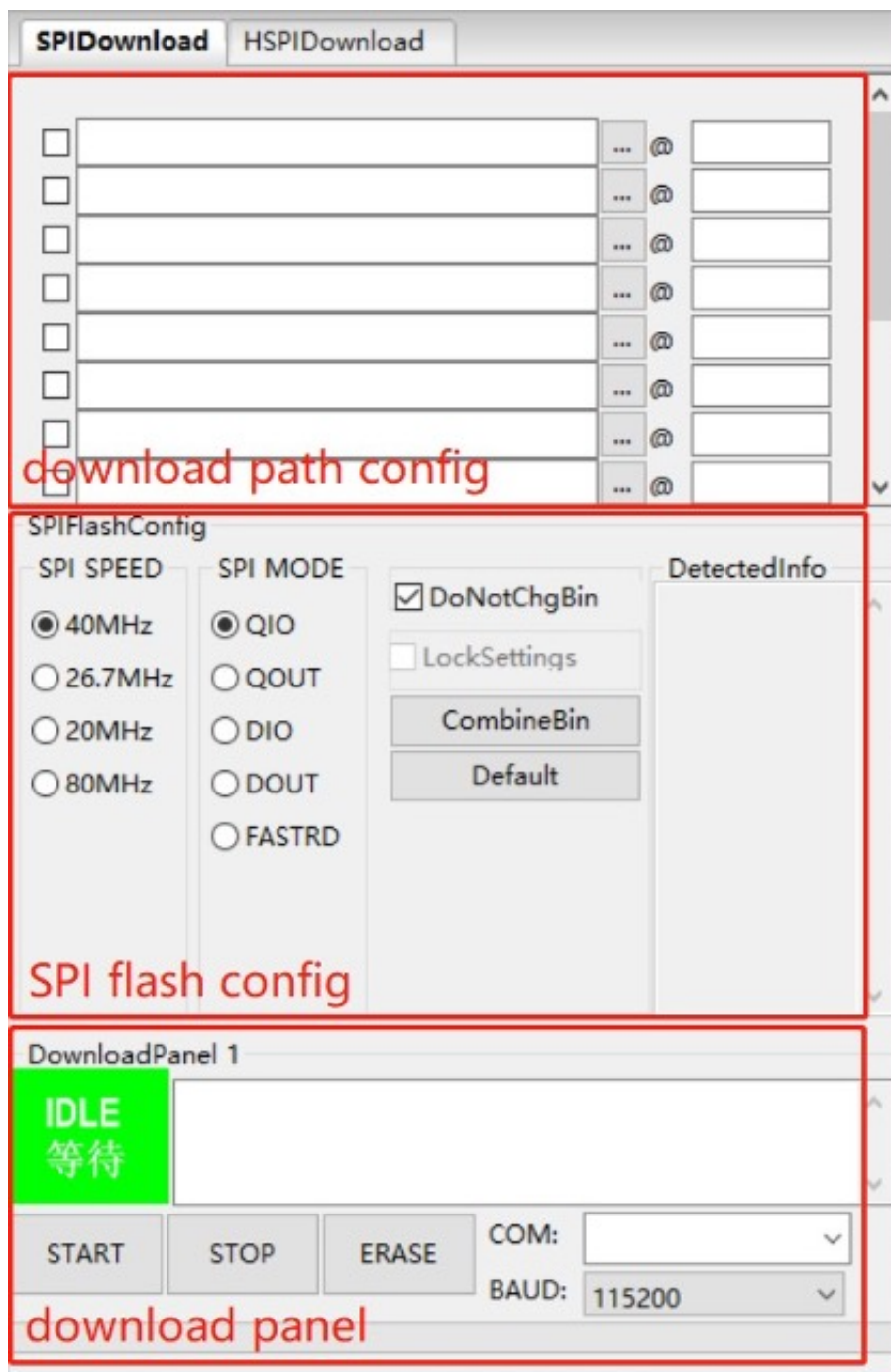


Fig. 62: SPIDownload Tab

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.)

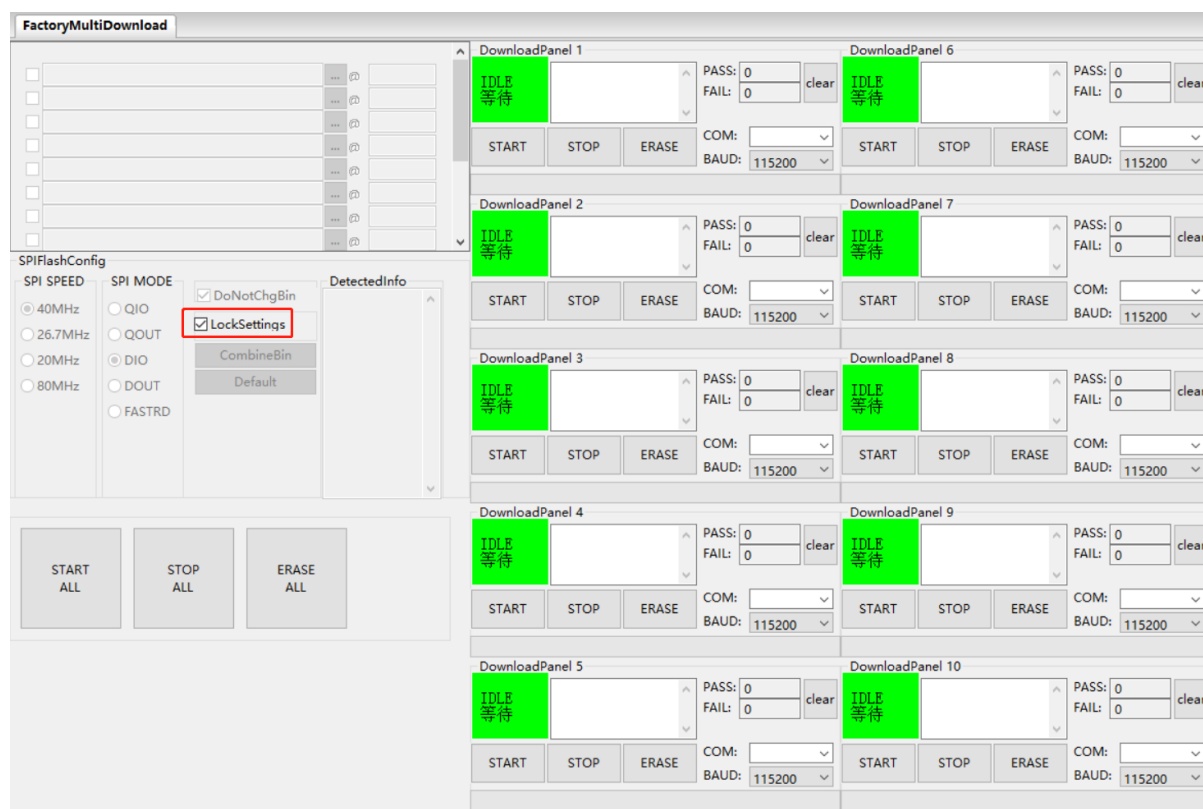


Fig. 63: FactoryMultiDownload Tab

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.

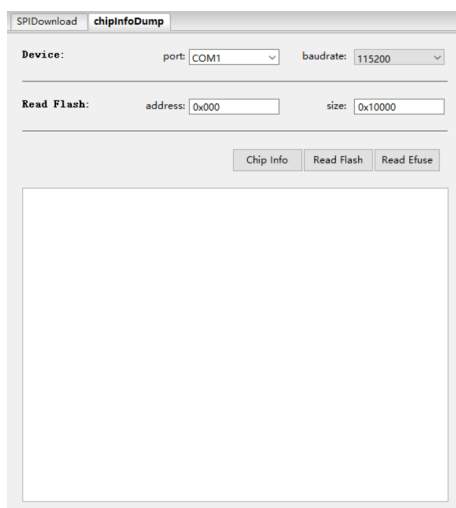


Fig. 64: chipInfoDump Tab

- **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 $\geq 3.9.8$
-

7.3 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-C61 only supports regular download, and will support encrypted download later.

Regular Download

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**

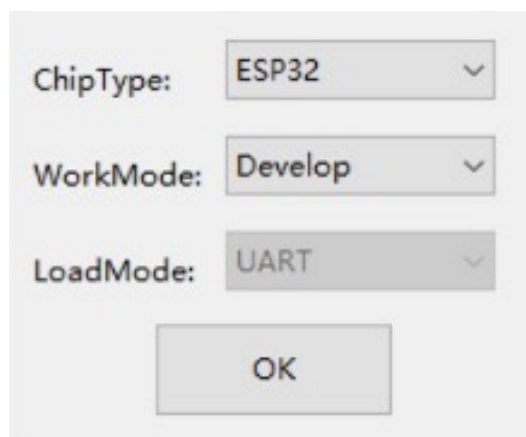


Fig. 65: Selecting Device —ESP32 Download Tool

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 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.

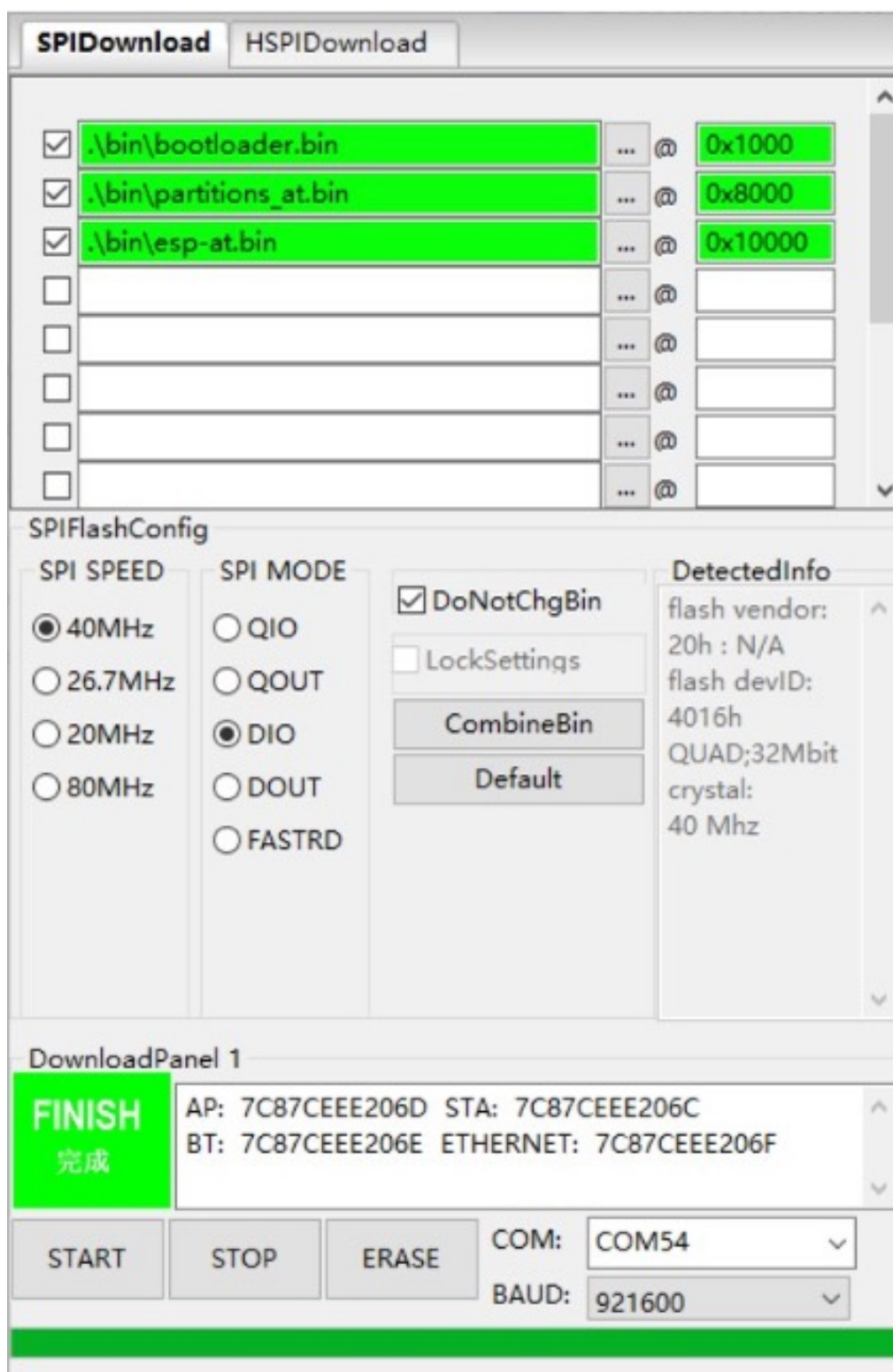


Fig. 66: Download Completed

Encrypted Download

The encrypted firmware downloading process is as follows:

- 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. 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:

- 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)
 - **public_key_digest_path = .securepublic_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)
- **[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 = .secureflash_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:

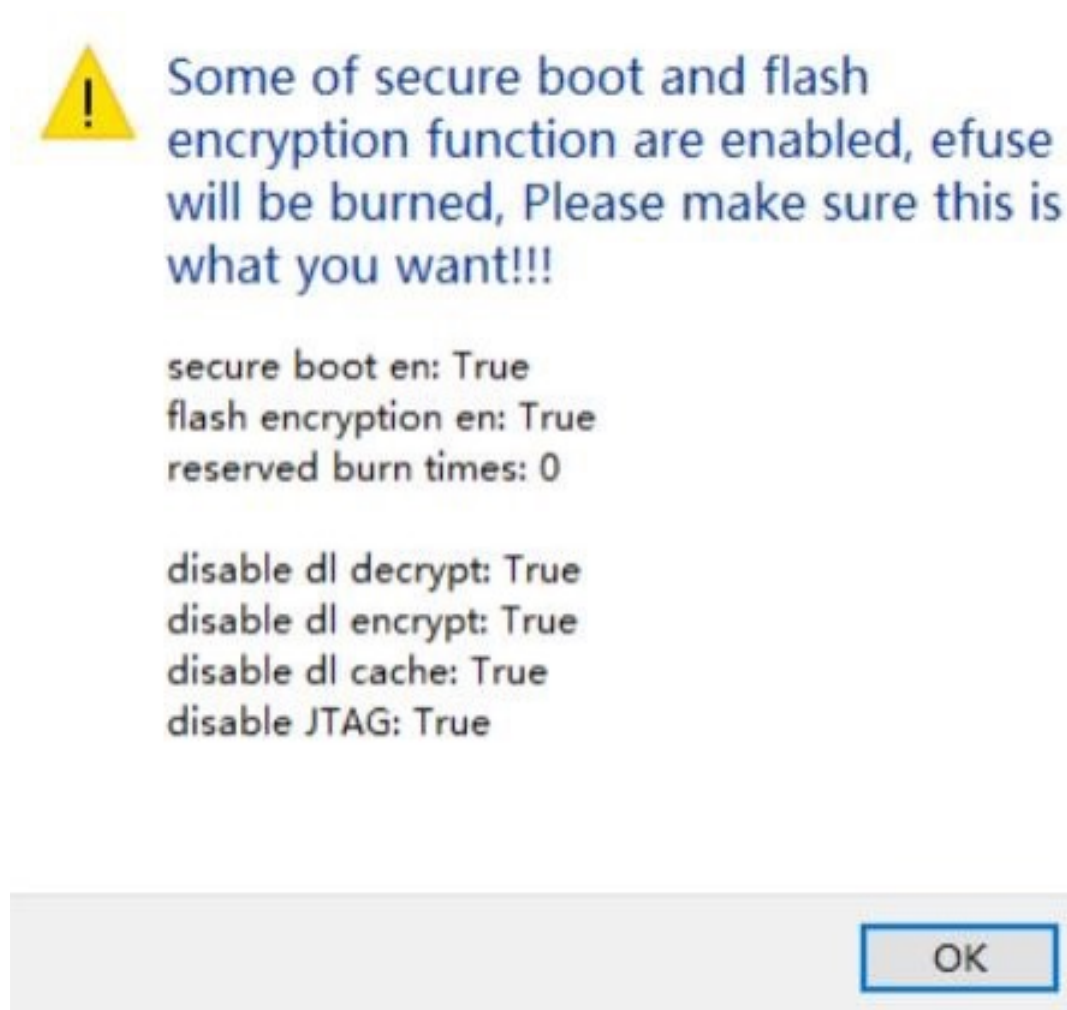


Fig. 67: ESP32 Prompt Message of Enabling Flash Encryption and Secure Boot

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.

Warning: This document is not updated for ESP32C61 yet, so some of the content may not be correct.
This warning was automatically inserted due to the source file being in the `add_warnings_pages` list.

8 Espressif Production Testing Guide

This guide mainly describes the production testing schemes available for Espressif Wi-Fi products (Wi-Fi module/Chip Onboard), thus providing reference for the production testing of customer products.

8.1 Introduction

Generally, there are two production testing schemes available to test the RF performance of the Wi-Fi products based on Espressif IC:

- RF General-purpose Tester Scheme (general standard in the industry);
- Signal Board Scheme (ESP enterprise standard).

RF General-purpose Tester Scheme

The tester scheme is widely used for the production testing of Wi-Fi products. Espressif provides the necessary serial port commands and firmware, so the customers can easily use this scheme for testing.

The testing steps can be found below, which are also demonstrated in the figure below:

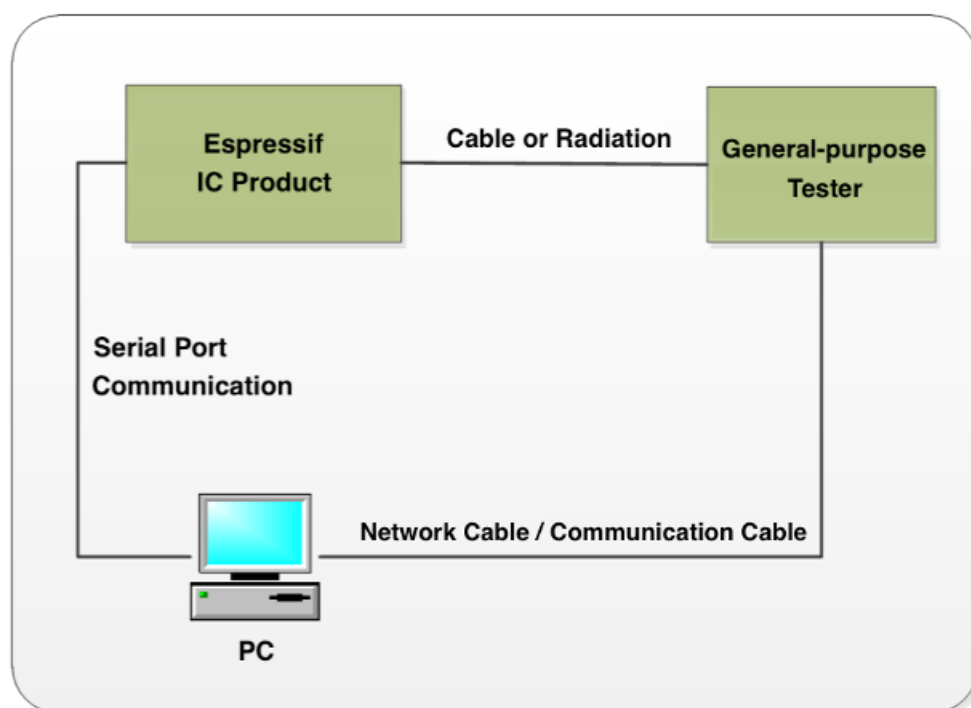


Fig. 68: Diagram of the Tester Scheme

1. Download `RF_Test_FW.bin` to ESP IC RAM;
2. Run the Test Tool that corresponds with the RF tester on the PC, and send the serial port commands to the modules for sending/receiving packets in different modes;
3. The RF tester analyzes the RF related parameters in each specific mode.

Note:

- For this scheme, the mass production testing tool provided by the tester supplier should be adaptable with the Espressif IC products;
- **If the supplier is unable to provide this kind of test tool, customers can alternatively make the tool adaptable by using the tester's available interfaces. Meanwhile, customers can use esptool to download `RF_Test_FW.bin`:**
 - esptool can be downloaded from [here](#). For related commands, refer to the [documentation](#);
 - For the manual testing of ESP products' RF performance, please refer to [RF Test Items](#).
- The test above must be performed in a shielded enclosure.

Signal Board Scheme

The signal board scheme is specially designed by Espressif, which can effectively test the RF performance of the mass-produced Wi-Fi products, and therefore guarantee the RF quality. This scheme features low cost of hardware and easy environment setup for factories.

As demonstrated in the figure below, the signal board can be used as a standard device to interact with and test the DUT (Device Under Test) by analyzing the communication data.

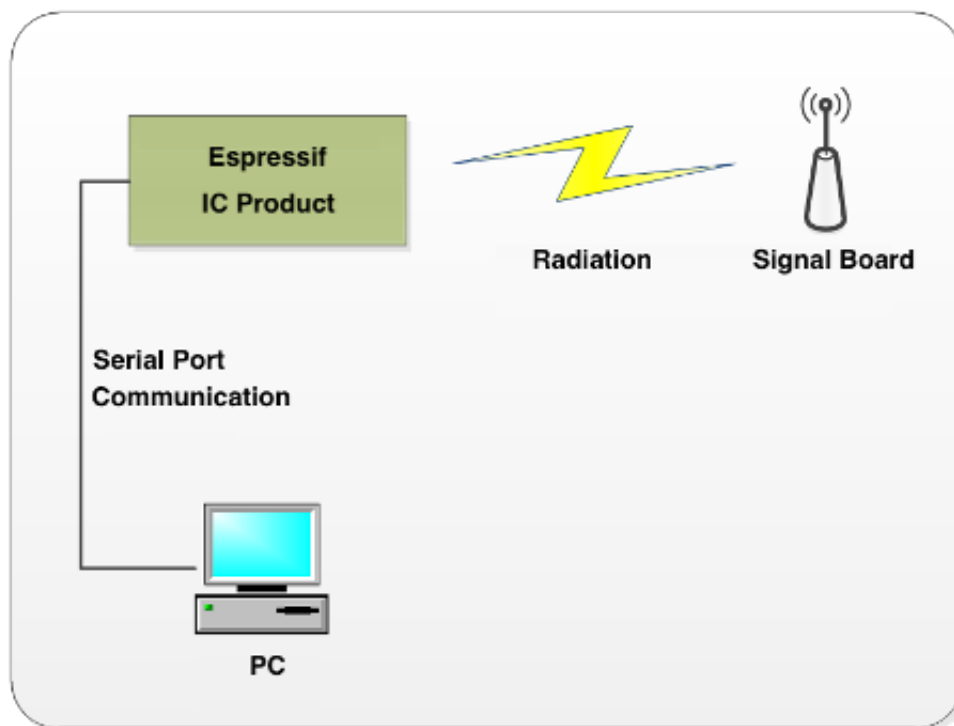


Fig. 69: Diagram of Signal Board Scheme

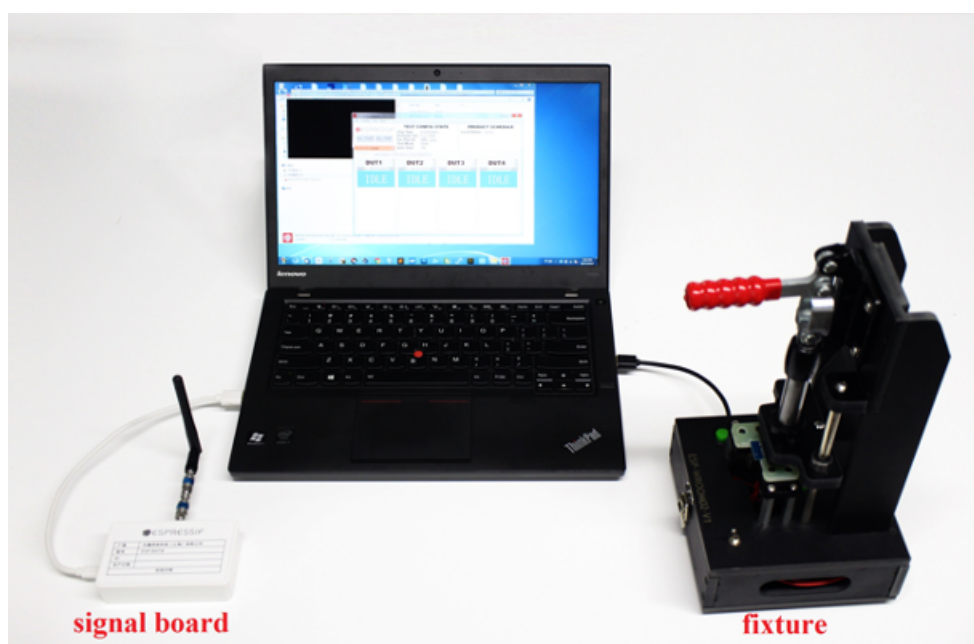


Fig. 70: Hardware Connection for Signal Board Scheme

Note: The test above must be performed in a shielded enclosure.

Production Testing Process

1. Please find the followings that need to be tested, and connect the DUT accordingly to conduct the test:

Note:

- Download mode: for downloading bin files and is the main mode used for production testing.
 - Flash operation mode: for checking the log info.
-

2. Connect the DUT to the serial port board by using the test fixture, and enter the DUT into the download mode via the production testing tool. If your serial port board does not support flow control, connect the corresponding GPIO(s) to low level directly, so DUT enters the download mode after powering up automatically.
3. Start the production testing tool on your PC, and follow the instruction provided in [Production Testing Tool](#).

Note:

- To enhance production efficiency, test fixtures are typically designed for one-to-multiple configurations. For instance, a common set-up is one-to-four, where a single fixture can accommodate and test four devices simultaneously.
 - For more details about the test fixture manufacturing instruction, please refer to [Test Fixture Manufacturing Instruction](#).
 - Connect the serial port board (which is placed inside the bottom box of the fixture) to the PC with a USB cable, and install the corresponding driver to ensure the serial port can be successfully identified.
-

Related Equipments for Production Testing

Serial Port Board The serial port board is mainly used as a USB converter. You may use other similar boards. However, considering some of them might have unstable performance, it is recommended to purchase what is shown below. If you want to purchase from Espressif, please [contact us](#).

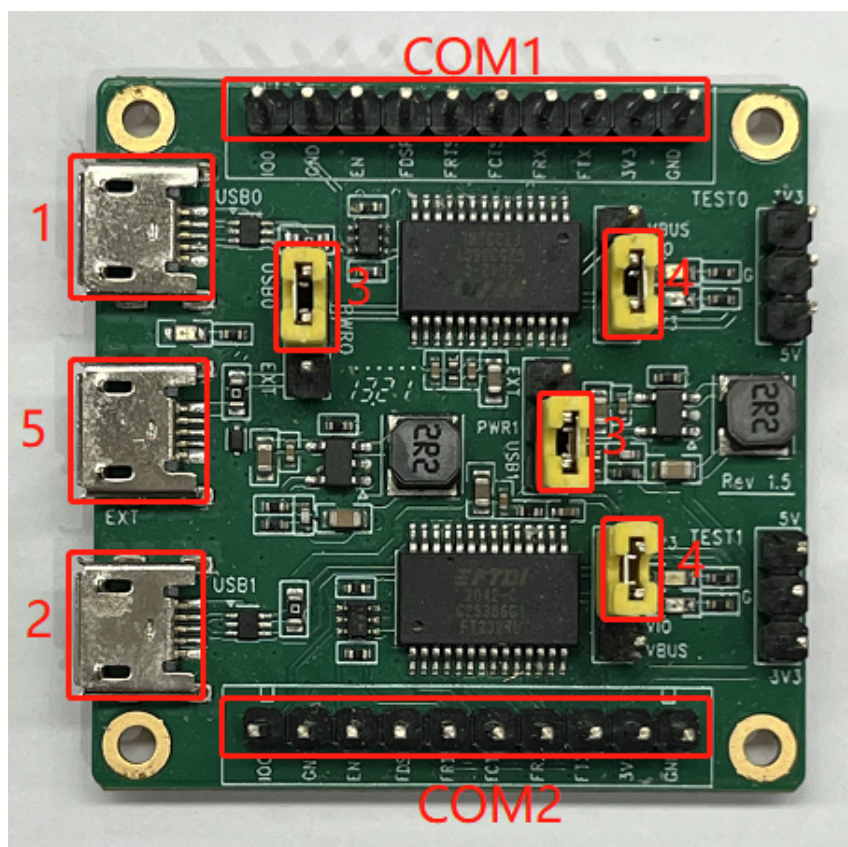


Fig. 71: Serial Port Board

Please check the board you purchased against the following requirements to make sure the switches and shorting jumpers are in the correct positions:

- Marking COM1 and Marking COM2: the serial ports used for the communication with the PC. Marking 1 and Marking 2 are the two independent serial ports, corresponding to TX/RX/FRTS/FCTS.
- Marking 3: selects powering the device through the USB port or an external power supply.
- Marking 4: selects 3.3 V or 5 V serial voltage level.
- Marking 5: for connecting external power supply, not used, so no need to configure.

Test Fixture The test fixture is an important equipment to execute the DUT in test mode. Specifically, you can put the module on the fixture and bring the module pins into contact with the fixture probes by pressing the fixture handle. When the test is finished, lift the handle to separate the module pins from the probes (for other similar equipments, customers may think of it as a reference or directly lead out the corresponding pins that are assigned to what you want to test).

For the production testing of Wi-Fi modules, the module pins must be led out and connected to the base board, so as to communicate with the serial ports of the PC. To achieve this, a fixture can be used. The figures below show the overall appearance of a typical fixture.

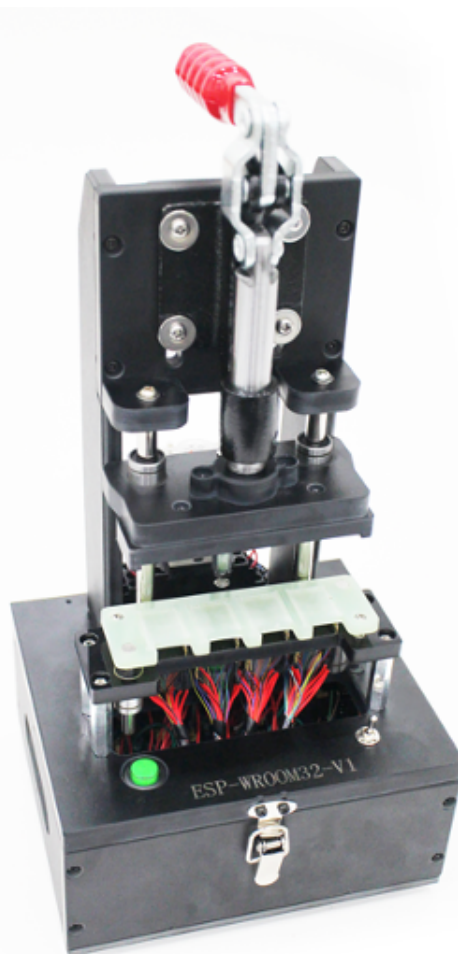


Fig. 72: A Typical Module Fixture

The primary structure of a typical module fixture can be seen in the table below (take Espressif test fixture as an example).

Table 6: The Primary Structure of a Typical Test Fixture

Part	Description
Handle	When users lift the handle, the module is separated from the metal probes at the bottom and gets disconnected from the power supply. When users press the handle, the module comes into contact with the metal probes and starts the testing procedure.
Mounting Panel	It is used for placing and holding the module.
Bottom box	It is used to place serial port board(s), enabling the module to communicate with the PC via USB.

Signal Board The signal board can be used a standard device to interact with the DUT during the production testing.

For the purchase of Espressif signal board, please [contact us](#).

Note:

- Only one signal board should be used within the same network coverage. Otherwise, signal interference will occur.

- If more than one signal board are used for mass testing, please conduct in a shielded room or with a shielded box.

Scheme Comparison

The comparison between the signal board scheme and the tester scheme is shown in the table below. You can choose from these two schemes according to your actual requirements.

Table 7: Scheme Comparison

Scheme	Test Item	Description
Signal Board Scheme	RF Test	Tests the supply voltage of the chip and its fluctuation, and the frequency offset against the signal board, etc.
	Packet Sending/Receiving Test	Tests the packet sending/receiving between the DUT and the signal board.
	GPIO Conductivity Test	Identifies IC soldering defects, if there are any.
	Firmware Version Verification Test	Verifies the version information of the firmware that has been downloaded to flash.
Tester Scheme	Flash RW Test	Verifies the RW operation of flash.
	EVM Test	Tests the TX Power, and EVM performance of the DUT during the packet sending.
	Frequency Offset Test	Tests the frequency of the DUT during the packet sending.
	TX Power Test	Tests the TX power of the DUT during the packet sending.
	RX Sensitivity Test	Tests RX sensitivity of the DUT (This test must be performed in an RF shielded environment).
	GPIO Conductivity Test	See above in this table.
	Flash RW Test	See above in this table.

Note:

1. The signal board scheme has applied Espressif's internal standards and can effectively ensure the quality of RF products, provided that the RF matching of the module is qualified and the production materials are consistent with those specified in the production processes.
2. To ensure the overall quality of the mass production of modules, the customers may use the signal board scheme for full inspection and the tester scheme for sampling inspection.
3. You cannot directly test the RF performance parameters of a DUT, such as TX, RX, EVM, and FREQ, with the signal board scheme. Therefore, a general-purpose Wi-Fi tester can be used as a supplement to the signal board scheme.

The signal board scheme features low cost and easy environment setup, making it a popular solution that has long been widely used by the customers. Therefore, this guide will mainly focus on this scheme, and demonstrate it with the use of Espressif modules. Customers may follow this guide to set up the testing environment for their own Wi-Fi products.

8.2 Environment Setup

In order to show the customers the effectiveness of our signal board scheme, Espressif provides a complete testing package. Our testing package introduces the customers to the overall process of the production testing. Note that an Espressif module is used in the package as an example, so that the customers can simply replace this module with their Wi-Fi products (of ESP32-C/ESP32/ESP32-S/ESP8266 series) in their own production testing.

Testing Package

The production testing package has the following key components:

Test Step

Please connect your DUT to the production testing base board as shown in the following figures, depending on the product series (ESP32-C/ESP32/ESP32-S/ESP8266) your DUT belongs to; then, connect the serial port board, signal board and your PC together; then, open the production testing software tool on your PC.

1. The DUT communicates with the signal board at a rate of around 1 to 2 M. Configure the DUT to download mode and power up the serial port board.



2. Open the production testing software tool and complete the corresponding configuration based on which product series your DUT belongs to. For details, please refer to Section [Tool Configuration](#) below.
3. Click START button to start the test. During the testing, Parameter `fb_rssi` in the Log must be kept at around -50.
4. Troubleshoot based on the test results.

8.3 Production Testing Tool

Tool Introduction

Download Link

Directory

- `factory_test_ui_tool`: the main directory
 - `factory_test_cus_v1.0.exe`: the executable file
 - `config`: the configuration files run by the tool
 - * `.sys_config/.bin`: stores the test bin files for different chips
 - * `.sys_config/.spec_file`: stores the threshold files for different chips
 - * `.sys_config/.sys_settings.conf`: configures the current test bin file and threshold file
 - `logs`: stores test logs for each DUT

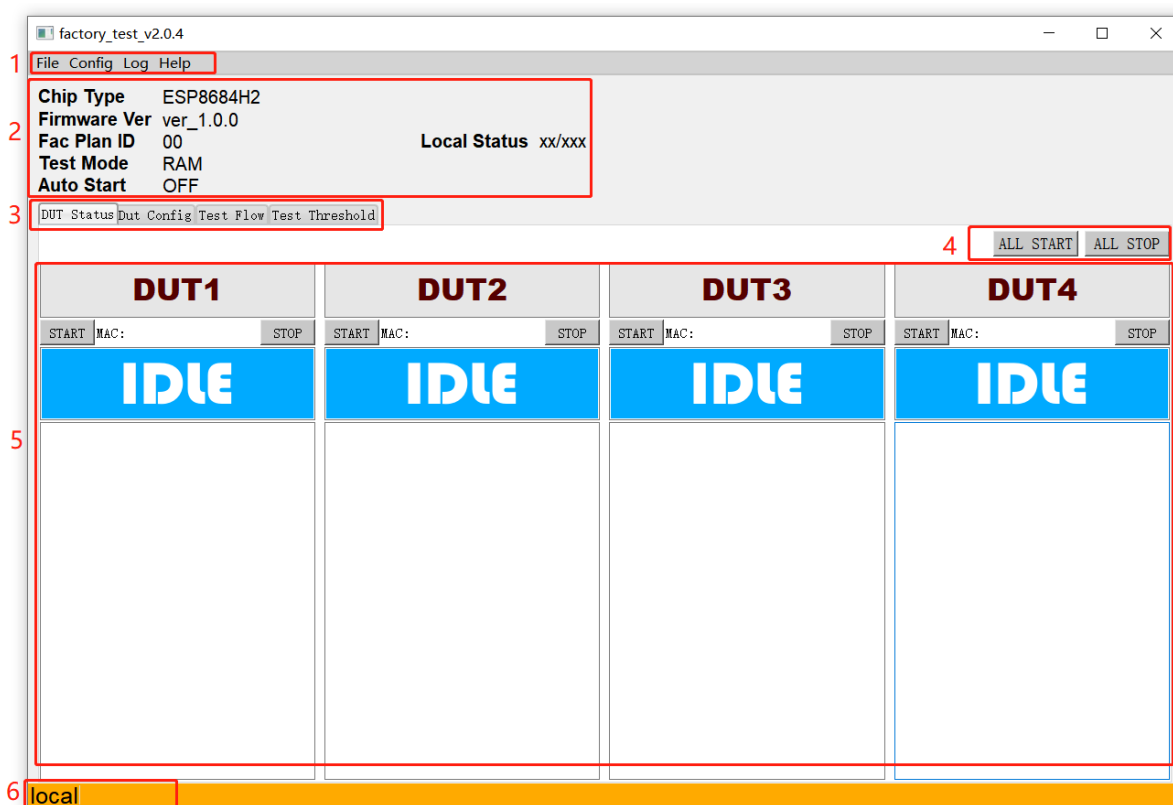


Fig. 73: Main Interface

Interface The MainWindows of the tool, as shown in the figure above, can be divided into six main parts:

1. Menu Bar:

- `Config` button can be used to switch between Local Mode and Cloud Mode (Cloud Mode is currently not supported);
- `Log` button can be used to select and open log files;
- `Help` button can be used to find help files.

2. Test Configuration Info:

- left section shows test configuration information such as Chip Type;
- right section shows the summary of all historical tests till now (the numbers of passed and failed tests).

3. Interface Tab Bar switches between different interfaces for testing or configuration.

4. ALL START/ALL STOP: Start/Stop all operations.

5. Testing Interface: the default testing interface after configuration. Here, you can see four DUT blocks, because a one-to-four fixture is used. Testing for different DUTs is independent from each other while the configuration of those is not.

6. Position displays if Local Mode or Cloud Mode is enabled.

Test Mode The current testing tool only supports 1 type of testing:

- **RAM Test:** Before testing, make sure the DUT is in download mode. During testing, the host computer downloads the firmware for testing to the RAM of the DUT, and runs it.

Tool Configuration

The screenshot shows the 'factory_test_v2.0.4' application window. At the top, there are tabs for 'File', 'Config', 'Log', and 'Help'. Below these, a status bar displays 'Chip Type: ESP8684H2', 'Firmware Ver: ver_1.0.0', 'Fac Plan ID: 00', 'Test Mode: RAM', 'Auto Start: OFF', and 'Local Status: xx/xxx'. The main area has four sub-tabs: 'DUT Status', 'Dut Config', 'Test Flow', and 'Test Threshold'. The 'Dut Config' tab is active and contains three distinct sections:

- Section 1 (TEST CONFIG):** Contains settings for 'chip type' (ESP8684H2), 'Test From' (RAM), 'Fac-Plan' (00), 'EFUSE MODE' (normal), and an 'AUTOST' checkbox.
- Section 2 (DUT1-DUT4):** A list of four DUT configurations. Each entry (DUT1, DUT2, DUT3, DUT4) has fields for 'PORT1', 'PORT2', 'Rate1', and 'Rate2'. For example, DUT1 is configured with PORT1: COM4, PORT2: COM4, Rate1: 115200, and Rate2: 115200.
- Section 3 (APPLY):** A large button labeled 'APPLY' used to save the configurations.

Fig. 74: DUT Config

Interface As shown in the figure above, the DUT Config tab can be divided into three major blocks:

- 1. EST CONFIG: Test-related configuration
- 2. DUT: DUT-related configuration
- 3. APPLY: apply the configuration

TEST CONFIG

Table 8: TEST CONFIG

Parameter	Description	Notes
Test From	Location from which the program starts to run	RAM: The test bin to be downloaded must be selected.
Fac-Plan	Test record code	The MAC list stored in the form of code + test result .
AUTOST	Automatic test switch	If this option is checked, a new test will start automatically when the current test finishes.
EFUSE MODE	Detection method of eFuse	If a customized MAC address is used, the <i>custom</i> option should be selected here. Otherwise, use <i>normal</i> .

DUT CONFIG

Table 9: DUT CONFIG

Parameter	Description	Notes
Port	Serial port number	Serial port configuration of the DUT, including the serial port for normal test and the serial port for firmware test. The configuration of these two serial ports can be the same or not, depending on the customers' actual requirements.
Rate	Baud rate	Baud rate of the serial port.
APPLY	Confirms all the modification that has been made	Any modification to the configuration on the interface takes effect only after clicking the APPLY button and passing the verification by entering the correct verification code. The verification code depends on the date and time of performing the test (the sum of the values of year, month, day and hour). For example, assuming the test is performed at 12 o' clock on March 1st, 2018, the verification code is the sum of 2018 (year), 03 (month), 01 (date) and 15 (hour), i.e. 2037.

Test Flow

In addition to some basic RF performance tests, the production testing scheme can also be used to evaluate the overall quality of the DUT through a variety of other tests, including but not limited to the RF Test (customizing commands is allowed), GPIO Conductivity Test, Firmware Version Verification Test, and Flash RW Test.

RF Test

- **Test Objective:** RF performance tests must be conducted during the production testing to ensure that the DUT can send/receive packets as expected.
- **Test Method:** Send/Receive packets back and forth between the signal board and the DUT through radiation. The DUT sends the test results to the host computer via the serial port. Then, the host computer provides conclusions by analyzing the test results.
- **Test Steps:** DUT runs with the testing firmware. The test starts after the host computer sends the serial port commands. Initially, the DUT obtains the RF performance information by reading registers, including the voltage fluctuations; then the DUT sends/receives a certain number of packets to/from the signal board; upon completion, the host computer determines if the DUT has passed the tests by checking the log information against the set thresholds.
- **Configuration Interface:** To enable this test, go to the `Test Flow` tab and check `RF_TEST`.

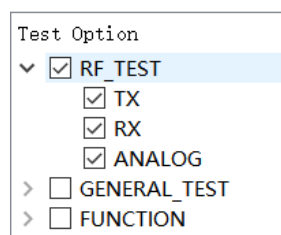


Fig. 75: RF Test

GPIO Conductivity Test

- **Test Objective:** This test can be performed to check the conductivity of GPIOs. It can help identify if there are any soldering problems, such as insufficient wetting or solder bridges.
- **Test Method:** After the corresponding pins of the fixture and DUT are shortened, the pin levels are set and obtained with the serial port commands, thus identifying any existing soldering problems.
- **Test Steps:** After the RF test is completed, a series of serial port commands are sent to perform the GPIO conductivity test. The serial port commands have been integrated in the host computer, so the customers can easily perform the GPIO Conductivity Test by enabling this function.
- **Configuration Interface:** As shown in the figure below, you can enable this test in the GENERAL_TEST sub-list on the Test Flow tab. For details, please refer to [Appendix B: GPIO Conductivity Test Configuration](#).

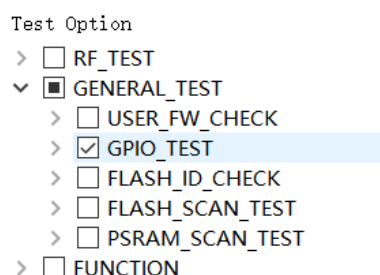


Fig. 76: GPIO Conductivity Test

Firmware Version Verification Test

- **Test Objective:** This test can be performed to verify the correctness of the firmware version downloaded to flash.
- **Test Method:** Check against the target firmware by comparing a “certain verification string” or “version number” in the serial port log. Therefore, this verification string must distinguish itself from those of other firmware.
- **Test Steps:** After the RF test, the host computer configures the serial port board to flow control mode, so the DUT boots from flash, and check the string in the log against the target string. You can easily verify the firmware version by enabling this function.
- **Configuration Interface:** You can enable this test in the GENERAL_TEST sub-list on the Test Flow tab.

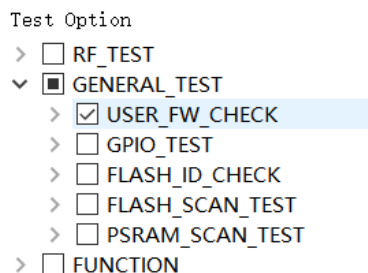


Fig. 77: Firmware Version Verification Test

Flash-related Test

- **Test Objective:** Flash ID Verification Test and Flash RW Test can be performed to verify the correctness of the flash model and the ability of the flash to read/write respectively.
- **Test Method:** These tests are performed with serial port commands, and the test results are returned to the host computer via the serial port.
- **Test Steps:** After the RF test is completed, the host computer provides a conclusion by checking the test results.
- **Configuration Interface:** You can enable these tests in the GENERAL_TEST sub-list on the Test Flow tab. FLASH_SCAN_ADDR is the starting address of this test item (only requiring a 0x1000 sector size), while FLASH_SCAN_TARGET is the target test value. The target value varies for different chips.

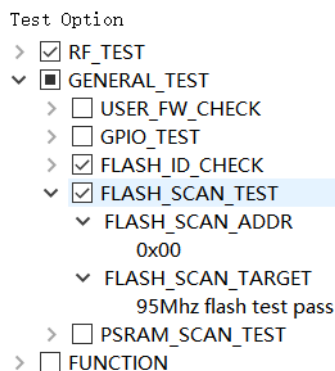


Fig. 78: Flash-related Test

Tool Operation

Two different test modes are supported in the signal board scheme: the Single-DUT mode and the Four-DUT mode. The configuration below is applicable to both of these two test modes. The operation process is as follows:

1. After setting up the environment, click **START** button (or **ALL START**) to begin synchronization and down-loading.

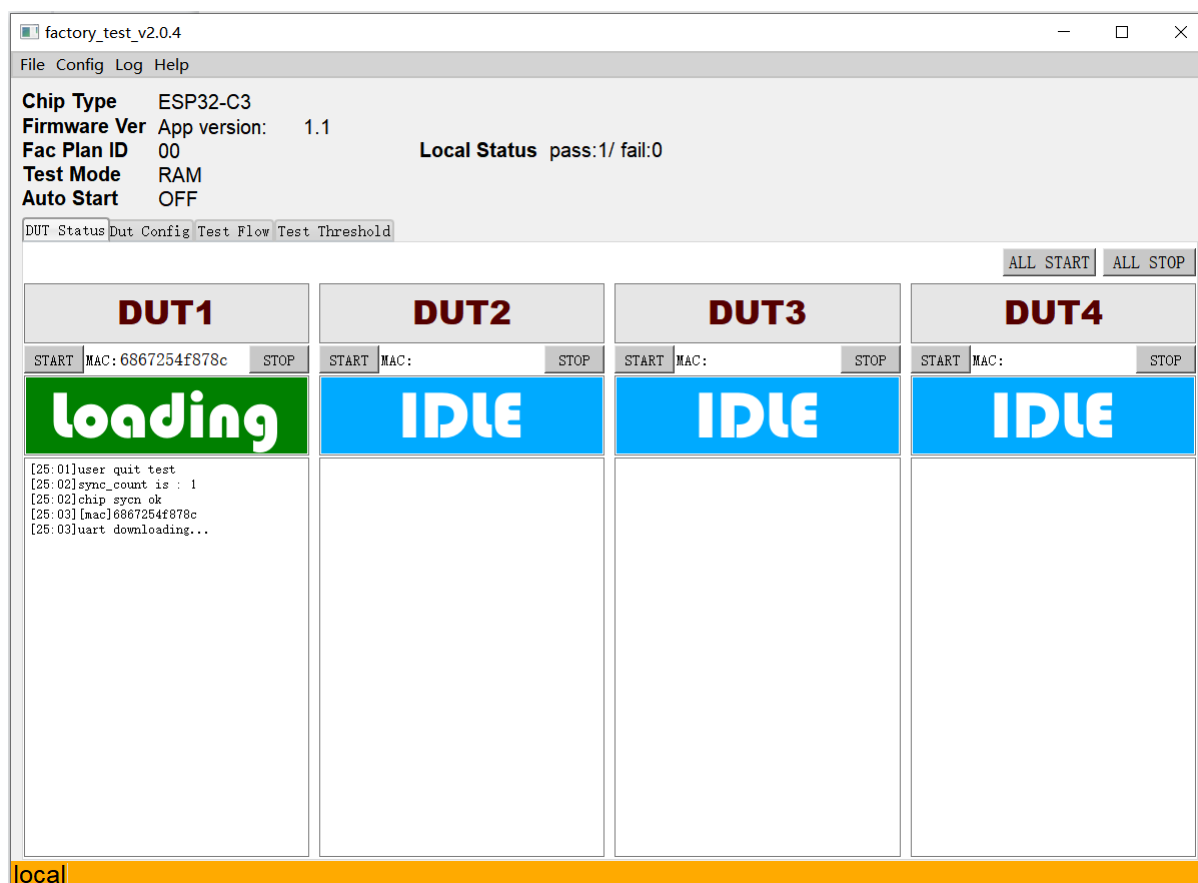


Fig. 79: Synchronization

2. After downloading is finished, the tool displays testing progress (RUN). Wait for test results.

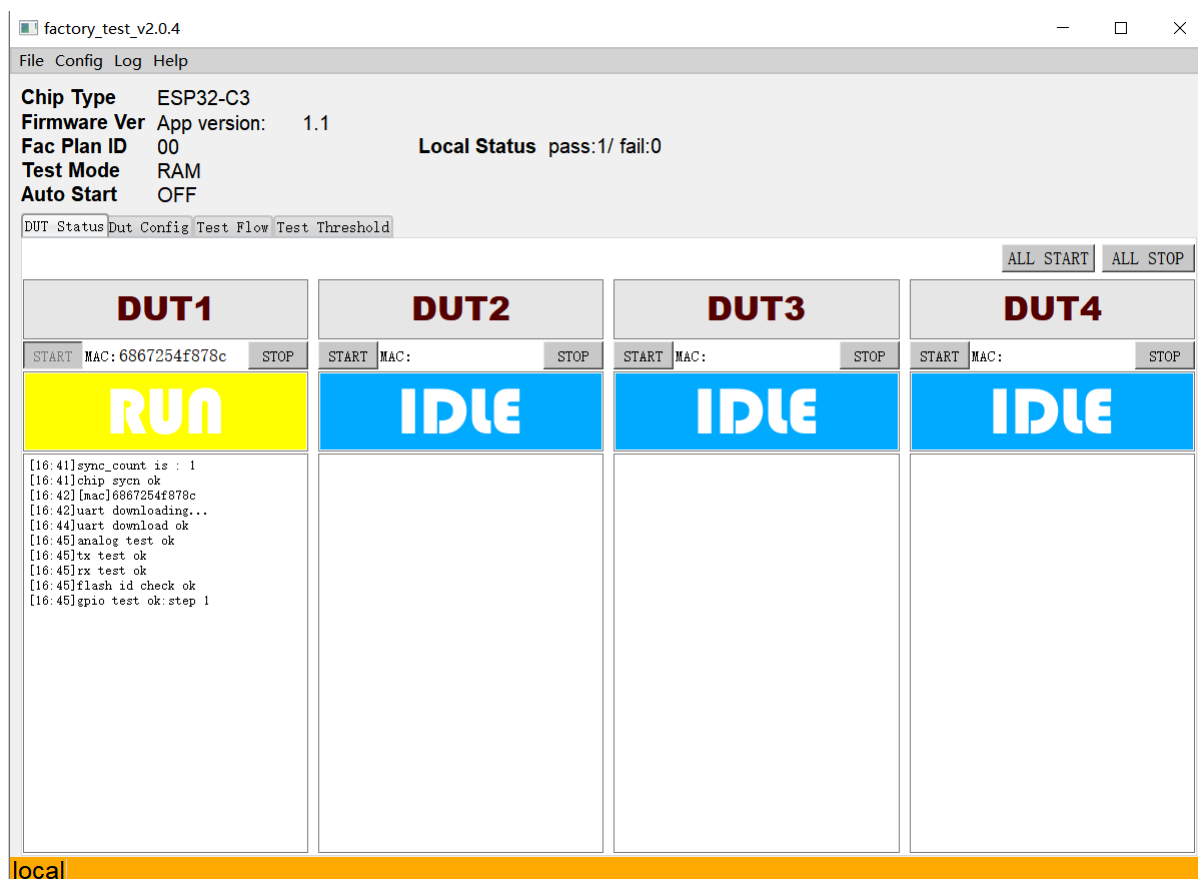


Fig. 80: Running

3. The tool displays test results.

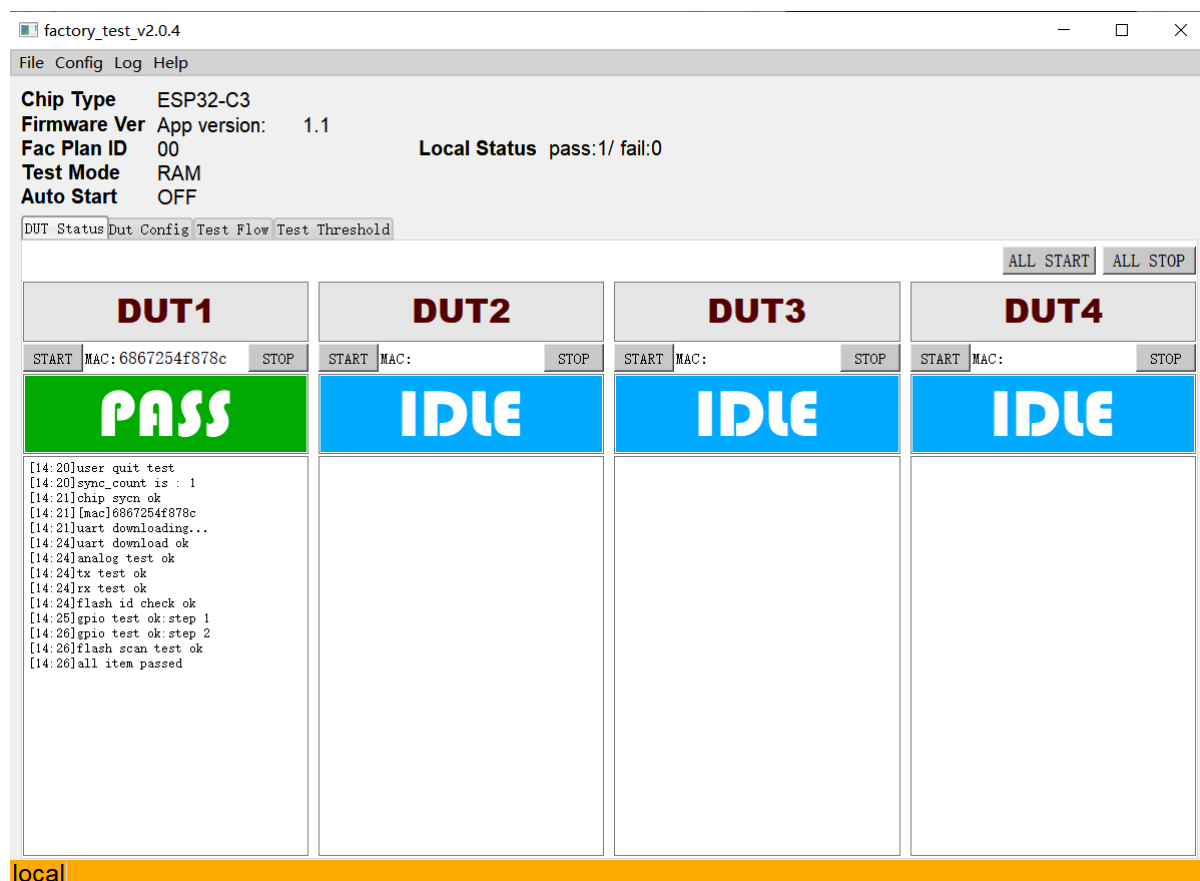


Fig. 81: Finish

The tool displays **FAIL** if any test item fails. The status block lists the detailed results of each test, to help you identify the reasons.

4. Check the detailed test records.

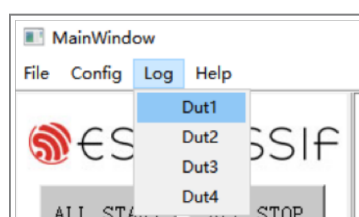


Fig. 82: Check Test Record

The test results of each production test will be saved in a separated log. The name of each log follows the pattern of “DUT MAC + date” . Click the **Log** button to bring up the log file of the last test for the corresponding DUT. If there is no test history for this workstation, open the **Logs** folder to access available logs.

8.4 Appendix B: GPIO Conductivity Test Configuration

During the GPIO Conductivity Test, the pins (GPIOx, GPIOy) to be tested should be connected with each other as instructed below. One pin works as an output for a signal ($n = 0$ or 1), while the other one works as an input and reads the current signal.

Note on <GPIOx, GPIOy, n>:

1. GPIOx is the input pin, and GPIOy is the output pin.

2. n can be 0 or 1. 0: low level; 1: high level.

To make sure both high and low levels of all the pins can be tested. Once configured, this GPIO conductivity test is always performed twice:

1. First time with the configured n, and
2. Second time with the inverted n.

For example, if n is configured to 1, then the test will run with n = 1 for the first time, and then run again with n = 0 for the second time.

8.5 Appendix C: Firmware Version Verification Test

The detailed configuration of the firmware version verification test is shown below:

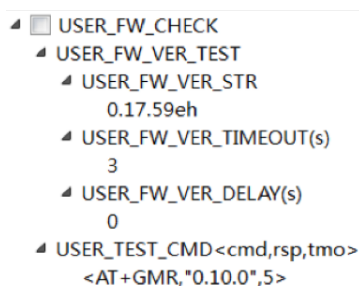


Fig. 83: Detailed Configuration

The firmware version verification test is enabled if the option `USER_FW_CHECK` is checked. Here, two test modes are available:

- `USER_FW_VER_TEST`: If the `USER_FW_VER_STR` is set to `Espcmd_en`, the `USER_TEST_CMD` is enabled. Otherwise, `USER_FW_VER_TEST` is enabled, in which a string is used to verify the firmware version number, such as the `0.17.59eh` in the figure above.
- `USER_TEST_CMD`: If the `USER_FW_VER_STR` is set to `Espcmd_en`, the `USER_TEST_CMD` is enabled. You can verify the firmware version number with your customized commands. For example, the value `<AT+GMR, "0.10.0" ,5>` in the figure above indicates that after sending the command `AT+GMR`, the serial port should return a character string that contains `0.10.0`.

8.6 Certification

Download certificates for Espressif products from [Certificates](#).

Warning: This document is not updated for ESP32C61 yet, so some of the content may not be correct.
This warning was automatically inserted due to the source file being in the `add_warnings_pages` list.

9 Test Fixture Manufacturing Instruction

9.1 About This Instruction

This document provides instructions on the manufacturing of the test fixtures for Espressif's Wi-Fi modules, in an effort to avoid problems caused by the lack of standardized fixtures during the module production and testing.

9.2 Overview

Module fixtures have different structures based on their types and usage. The structure of an ESP-WROVER fixture is shown in the following figure:

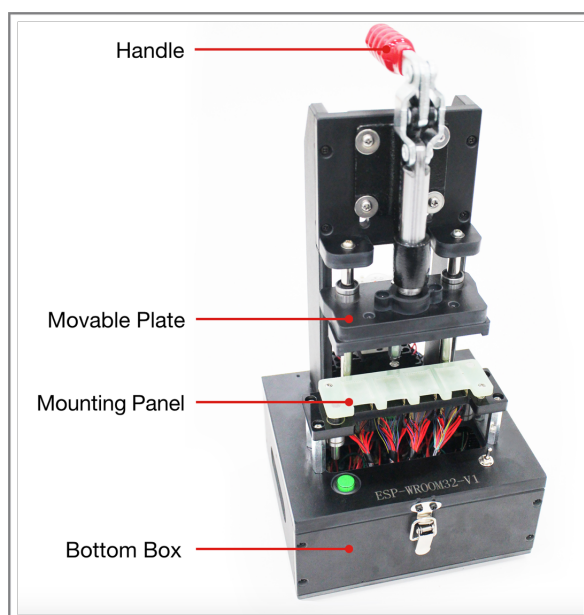


Fig. 84: The Structure of a Typical Test Fixture (ESP-WROVER)

The structure of other module fixtures are similar to that of ESP-WROVER. The primary structure of a typical module fixture consists of the following parts, which may differ only on the details:

Table 10: The Primary Structure of a Typical Test Fixture

Part	Description
Handle	It is used to power on or power off the module: - When users lift the handle, the module is separated from the metal probes at the bottom and gets disconnected from the power supply. - When users press the handle, the module comes into contact with the metal probes and starts the testing procedure.
Mounting Panel	It is used for placing and holding the module.
Bottom box	It is used to place serial port board(s), enabling the module to communicate with the PC
Switch	It is installed on the bottom box to control the power supply to the serial port board and the working modes.

9.3 The Main Structure of a Typical Module Fixture

Mounting Panel

The items needing attention during the manufacturing of mounting panels are listed below.

Antenna The antenna area should be completely exposed, keeping the antenna connection point over the line at the left end of the mounting panel or aligned with the left end of the mounting panel, which can be seen in the figure below. The mounting panel should not be made of metal, and the use of metal components should be minimized around the antenna:

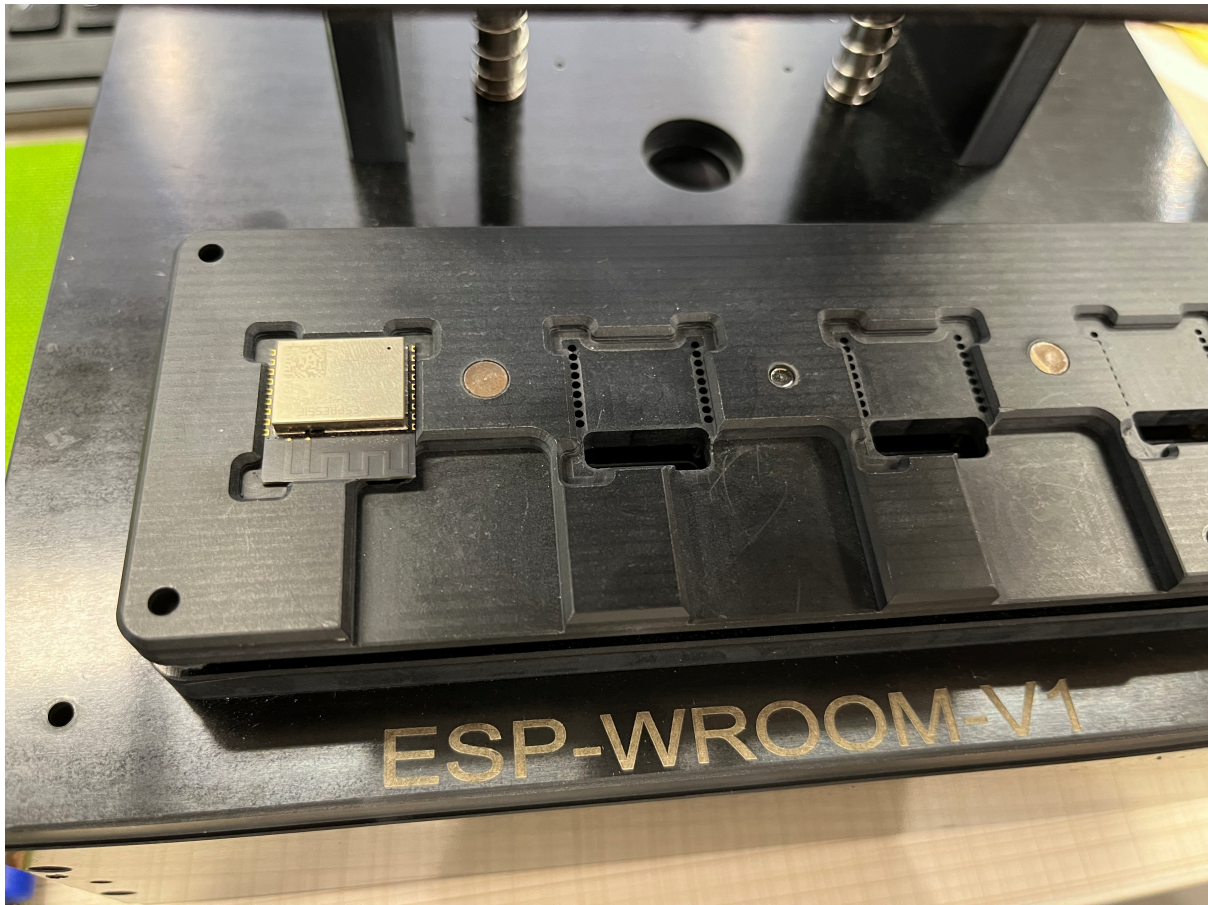


Fig. 85: The Mounting Panel for the ESP-WROVER Test Fixture

Handle

- When users press the handle, they must ensure that the metal probes under the mounting panel are attached to all the pins of the module.



Fig. 86: The Metal Probes (the handle is pressed)

- When users lift the handle, they must ensure that the metal probes under the mounting panel are completely detached from all the pins of the module, which can be seen in the figure below:



Fig. 87: The Metal Probes (the handle is lifted)

When users press the handle, they should leave a suitable distance between the movable plate and the mounting panel. The aim is to ensure that the probes are in touch with all the pins of the module, yet without crushing the module and its shield cover. Please see the figure below:

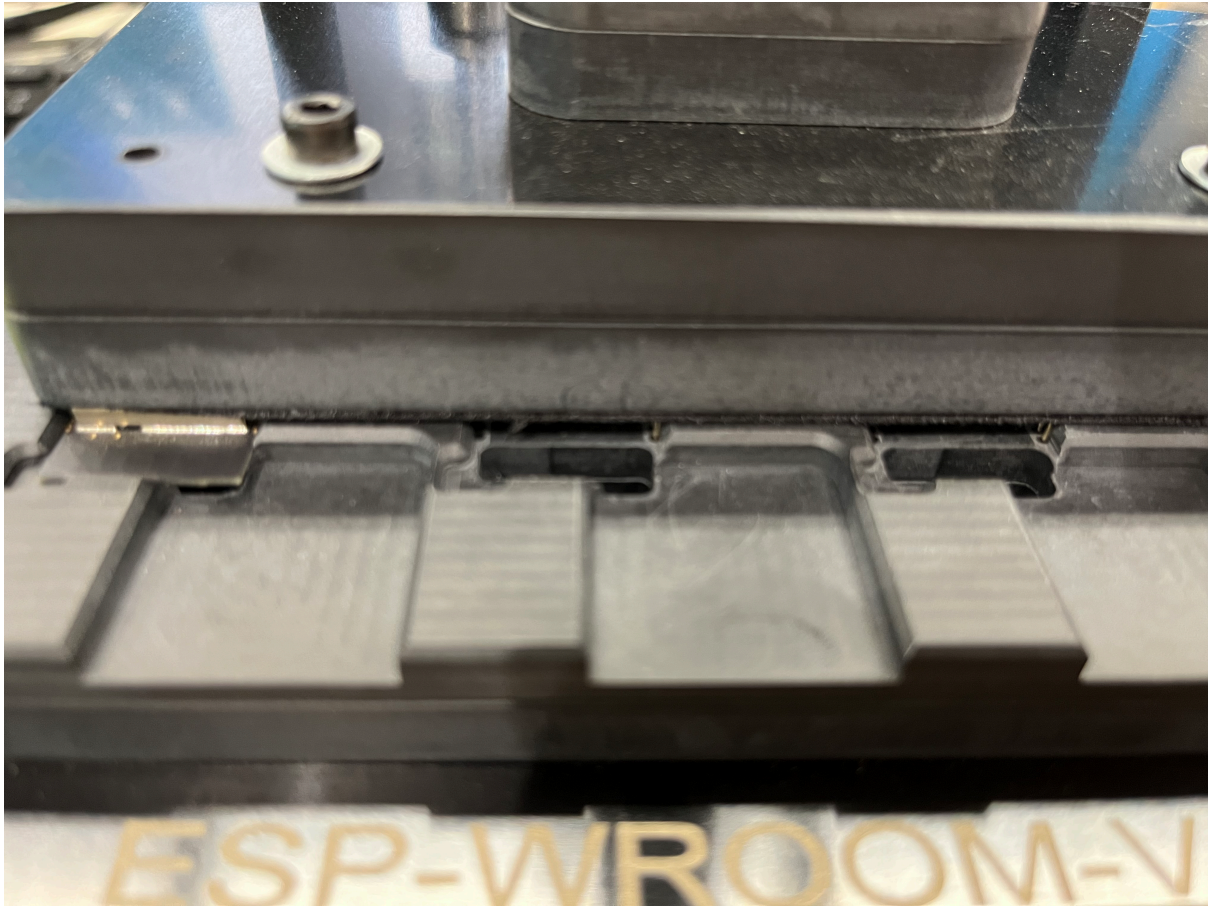


Fig. 88: The Movable Plate and the Mounting Panel

Bottom Box

Serial Port Board Two serial port boards (ESP_Factory Test Boards V1.3) are placed inside the bottom box, which can be seen in the figure below:

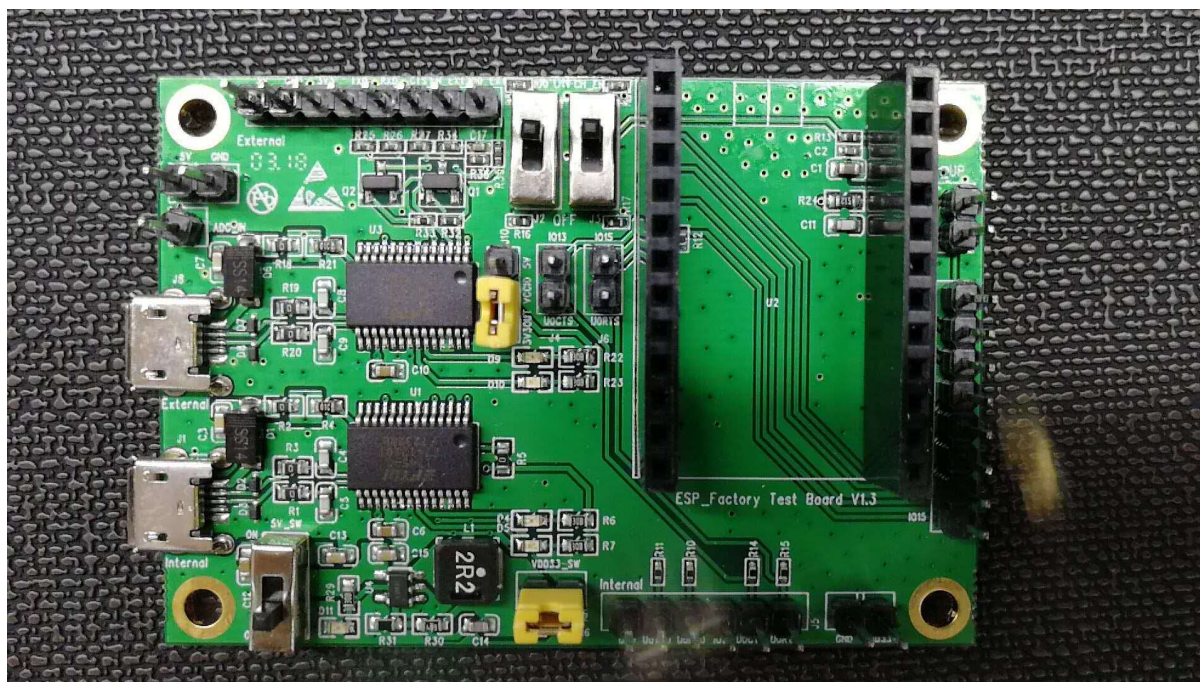


Fig. 89: A Typical Serial Port Board (ESP_Factory Test Board V1.3)

This serial port board, which has two serial-port chips, is placed inside the bottom box. Users should place as many serial port boards as necessary for the specific type of module fixture in operation. For example, the one-to-four module fixture requires four serial port boards.

It is required that the serial port boards be screwed onto the bottom box so as to keep them stable and prevent any short-circuit in the boards. The serial port boards are fixed by using screws in the four pass-through holes of the boards. In addition, when multiple serial ports are used to connect the HUB, the HUB should be provided with an external power supply to avoid a series of problems caused by insufficient power supply to the serial ports.

The Mark on the Bottom Box To facilitate the identification of the fixtures, it is required that the logo be printed on the surface of the bottom box. The classification of such marks is shown in the table below, where V* indicates the version of the fixture.

Table 11: The classification of the marks

Module Type	Mark
• ESP-WROOM-02 • ESP-WROOM-02D • ESP-WROOM-02DC	ESP-WROOM-02/02D-V1
• ESP-WROOM-02U • ESP-WROOM-02UC	ESP-WROOM-02U-V3*
• ESP32-WROOM-32 • ESP32-WROOM-32D • ESP32-WROOM-32DC • ESP32-SOLO-1 • ESP32-SOLO-1C	ESP32-WROOM-32/32D-V1
• ESP32-WROOM-32U • ESP32-WROOM-32UC	PESP32-WROOM-32U-V3*
• ESP32-WROVER (PCB) • ESP32-WROVER-B (PCB) • ESP32-WROVER-BC (PCB)	ESP32-WROVER-V1
• ESP32-WROVER (IPEX) • ESP32-WROVER-B (IPEX) • ESP32-WROVER-BC (IPEX)	ESP32-WROVER-I-V2*

Note:

1. Jumper caps in *The Movable Plate and the Mounting Panel* should be plugged in the yellow shorting plugs.
2. This guide is not applicable to ESP-WROOM-02U-V3, ESP32-WROOM-32U-V3 and ESP32-WROVER-I-V2.

The Wiring of the Mounting Panel Please see the rules in the table below, and connect the red and green wires as requested, while leaving the rest of the wires unconnected.

Table 12: Wiring Requirements

Components	Functions	Requirements	Remarks
Red wire	The red wires coming out of the probe	Connect the DuPont cables coming out of the probe to the serial boards. The pins with the same identification numbers should be connected to one another (see the following figures).	Notices : 1. Users should use standard DuPont cables, and the length of the cables should be kept as short as possible. 2. For the pins on the serial board that are not led-out, please solder the cables directly to the tin spots at the back of the serial boards.
	The red wires coming out of the switch	Connect the DuPont cables coming out of the switch to the serial boards. The pins with the same identification numbers should be connected to one another (see the following figures).	
Green wire	—	No need to connect the green wires to the serial port board.	—
Switch	One to one	One-channel side switch	—
	One to four	A four-channel side switch that controls four circuits.	—

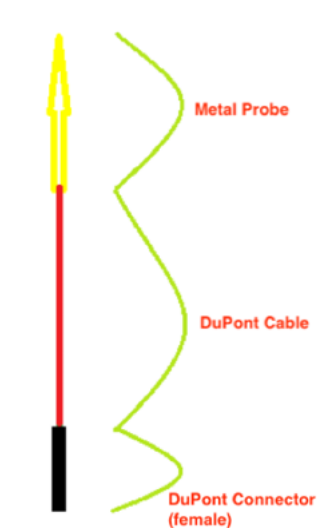


Fig. 90: A Diagram Showing How to Connect the Red Wires

The Wiring of the Fixture The fixture can enable or disable the **Automatic Mode Switching** on the Tool Side, by using the different wirings which can be seen in the following figures.

Note: By default, the Automatic Mode Switching on the Tool Side is not enabled.

When the Automatic Mode Switching on the Tool Side is not supported

1. ESP-WROOM-02

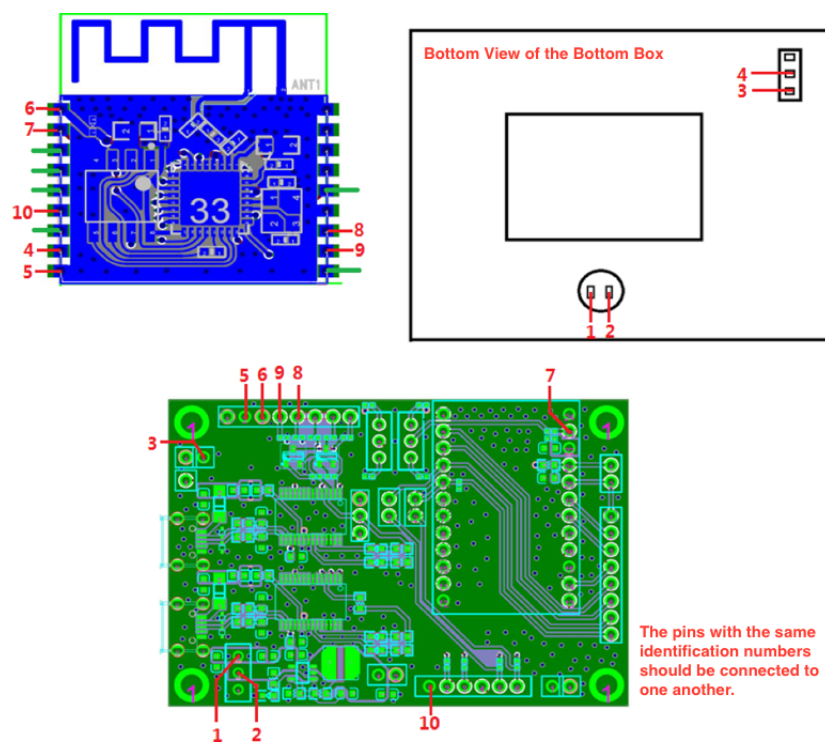


Fig. 91: The Wiring of the ESP-WROOM-02 Fixture

2. ESP-WROOM-32

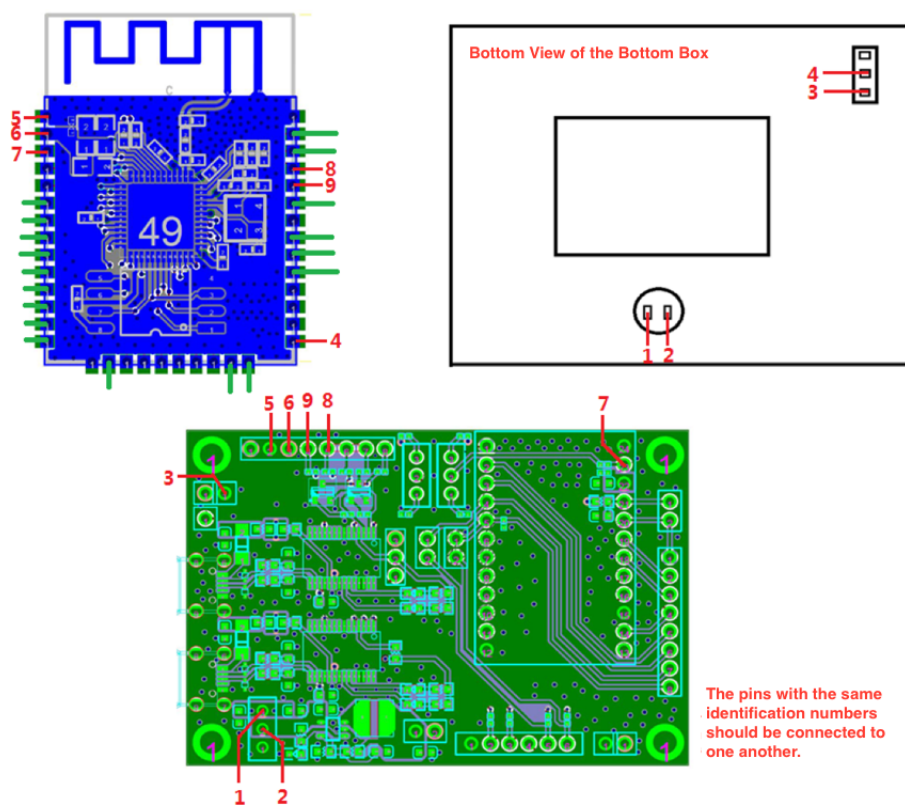


Fig. 92: The Wiring of the ESP32-WROOM-32 Fixture

3. ESP-WROVER

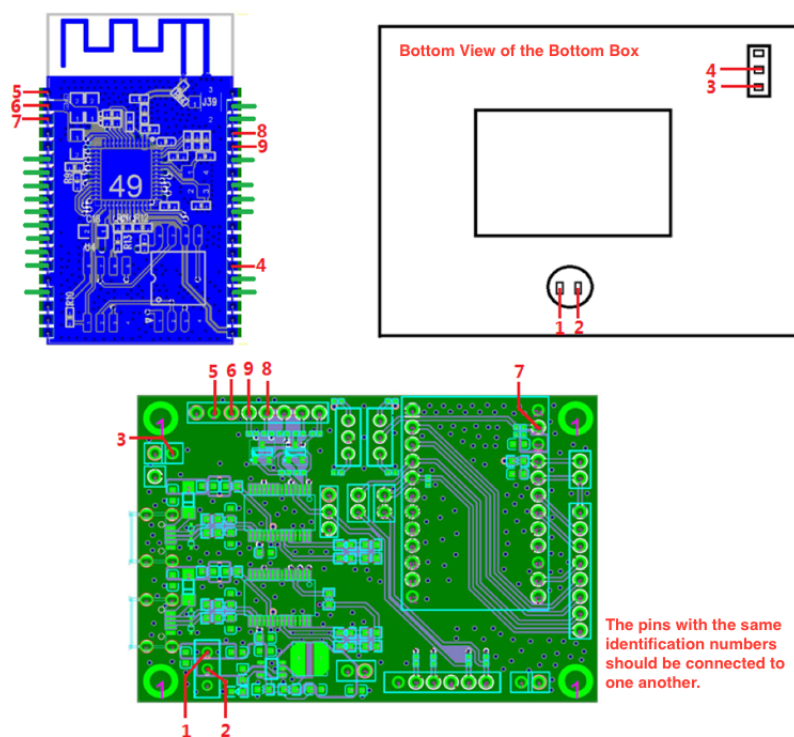


Fig. 93: The Wiring of the ESP-WROVER Fixture

When the Automatic Mode Switching on the Tool Side is supported

1. ESP-WROOM-02

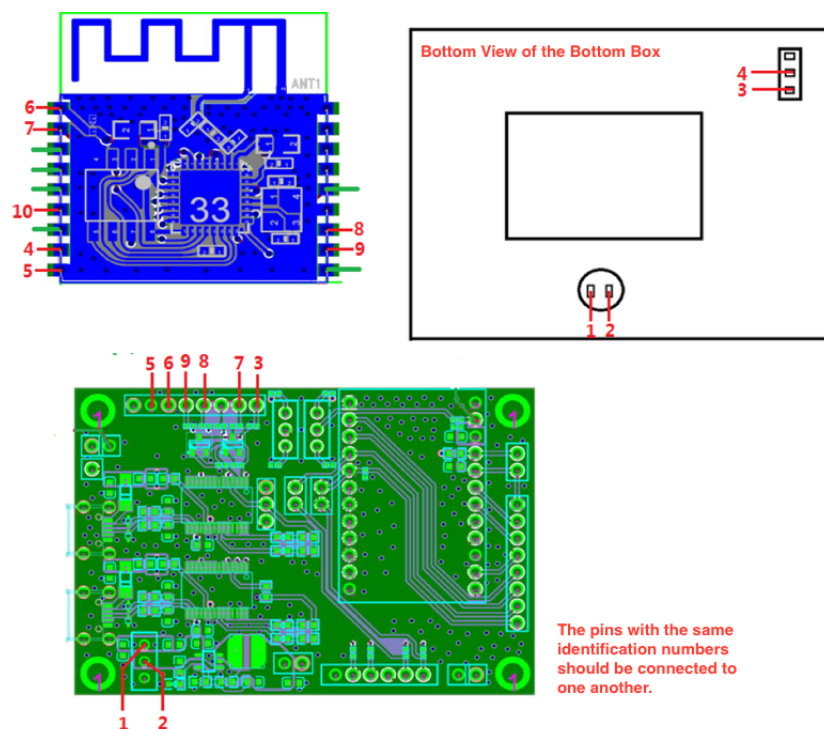


Fig. 94: The Wiring of the ESP-WROOM-02 Fixture

2. ESP32-WROOM-32

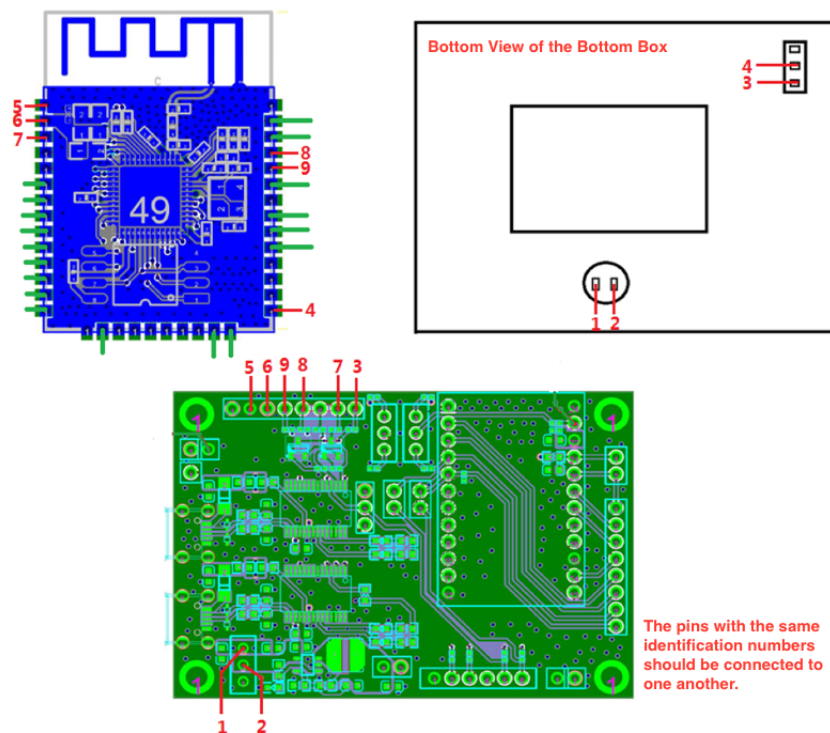


Fig. 95: The Wiring of the ESP-WROOM-32 Fixture

3. ESP-WROVER

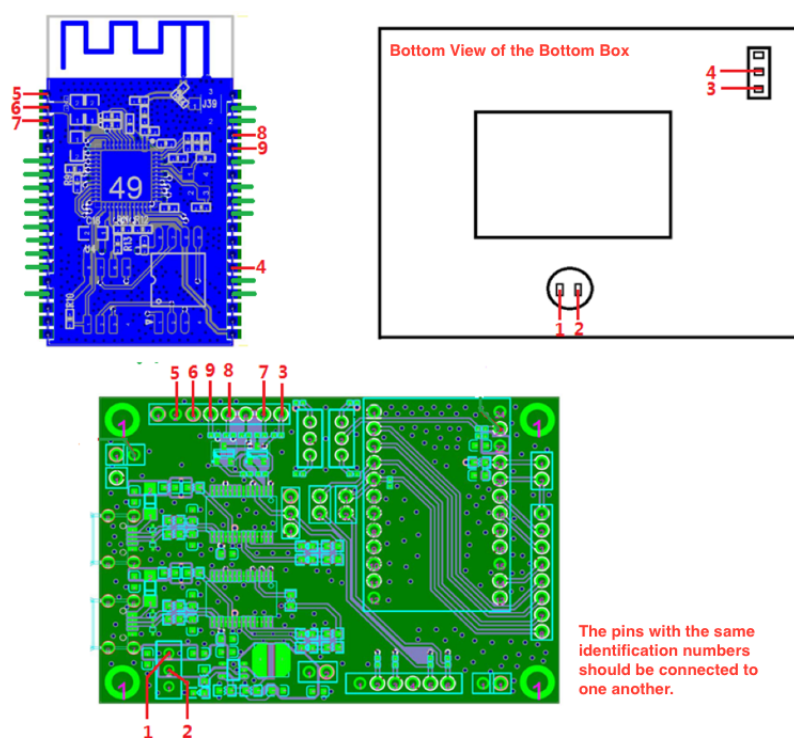


Fig. 96: The Wiring of the ESP-WROVER Fixture

9.4 Fixture Testing

Wiring Conductivity Test

To ensure that all the materials used for the wiring are functional, a conductivity test should be performed after the wiring is completed. Users can choose test tools, such as a multimeter, a simple LED circuit and so on.

Working Mode Verification Test

Please follow the steps below to verify the working modes of the module.

Operation Mode (currently only for ESP-WROOM-02 series modules)

1. After the wiring conductivity test, open a communication software for serial ports on the PC. (The **Serial Port Utility** is recommended here).
2. Select the corresponding port and baud rate (ESP8266/ESP32: 115200), and start the communication.
3. Use the switch located in positions 3 and 4 and toggle it towards position 3, so that you configure the module to the operation mode.
4. Press the handle.
5. Enter the command **AT+GMR**, and click Send.

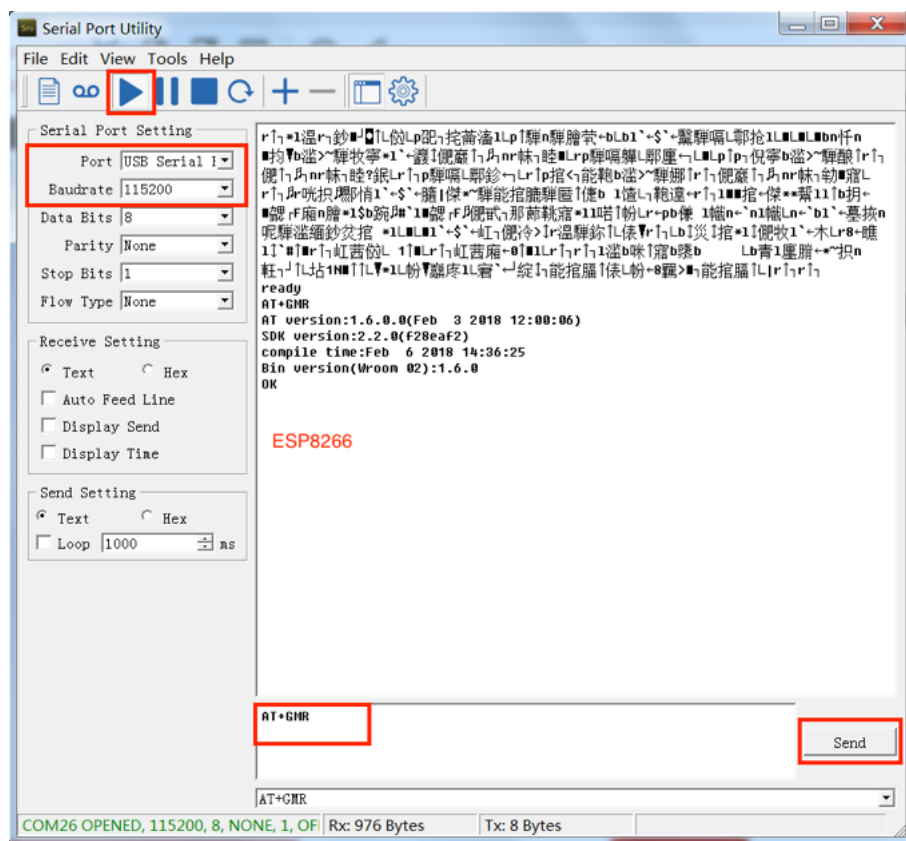


Fig. 97: AT Command Test

Note: Before clicking the Send button, users should press the **Enter** button after typing in the **AT+GMR** command.

6. Check the serial debugging tool window.

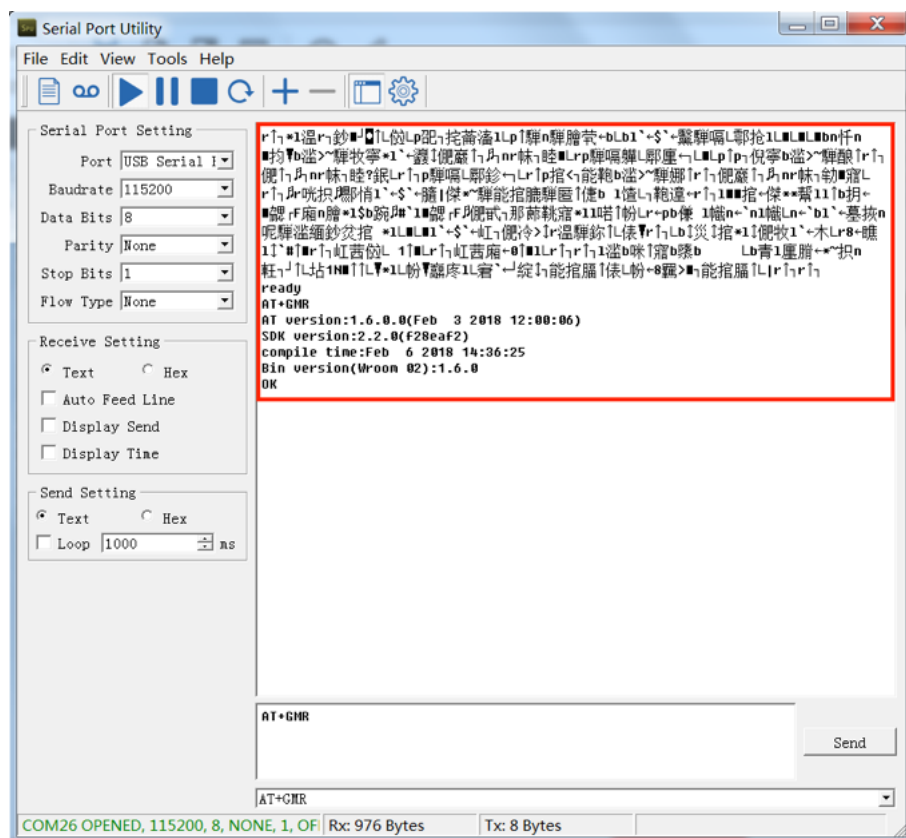


Fig. 98: Expected Result of AT Test - ESP8266 Series

- If the test result is as expected, which means the test was successful, users should save a screenshot of the test result.
- If the test result is not as expected, which means the test was not successful, users should perform the conductivity test again and ensure that the wiring is correct.

Download Mode

1. After the verification test of the operation mode, users should perform a verification test of the download mode, using the same serial port debugging tool.
2. Select the corresponding port and baud rate (ESP8266: 74880; ESP32: 115200), and start the debugging.
3. Use the switch located in positions 3 and 4 and toggle it towards position 4, so that you configure the module to the operation mode.
4. Press the handle.
5. Check the serial debugging tool window.

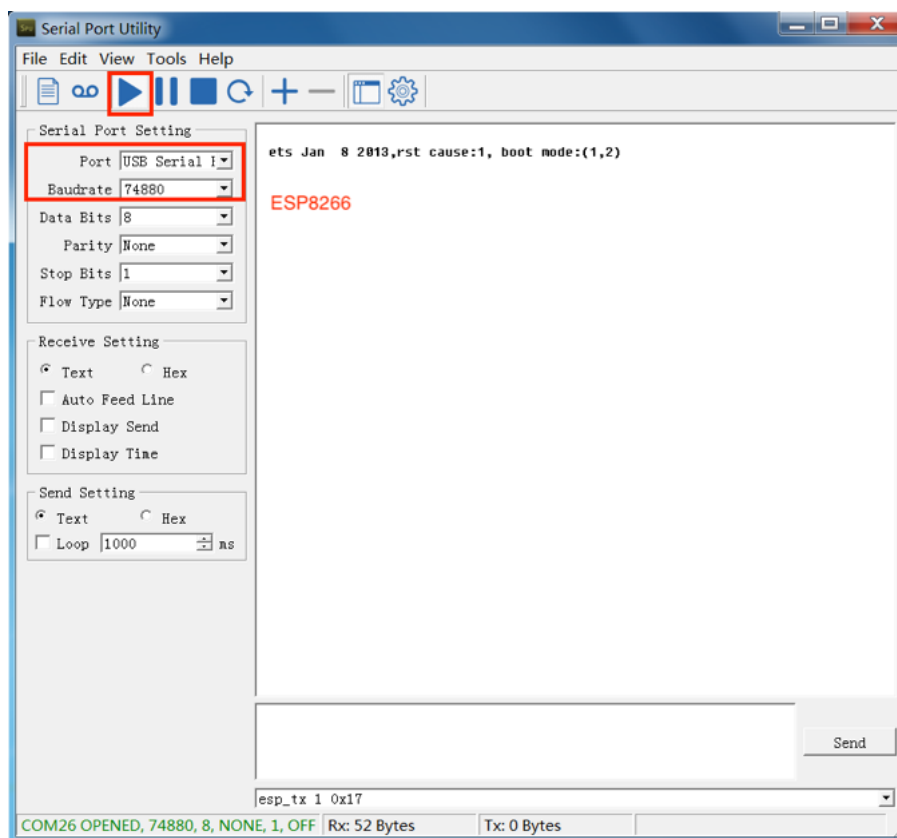


Fig. 99: Expected Result - ESP8266 Series

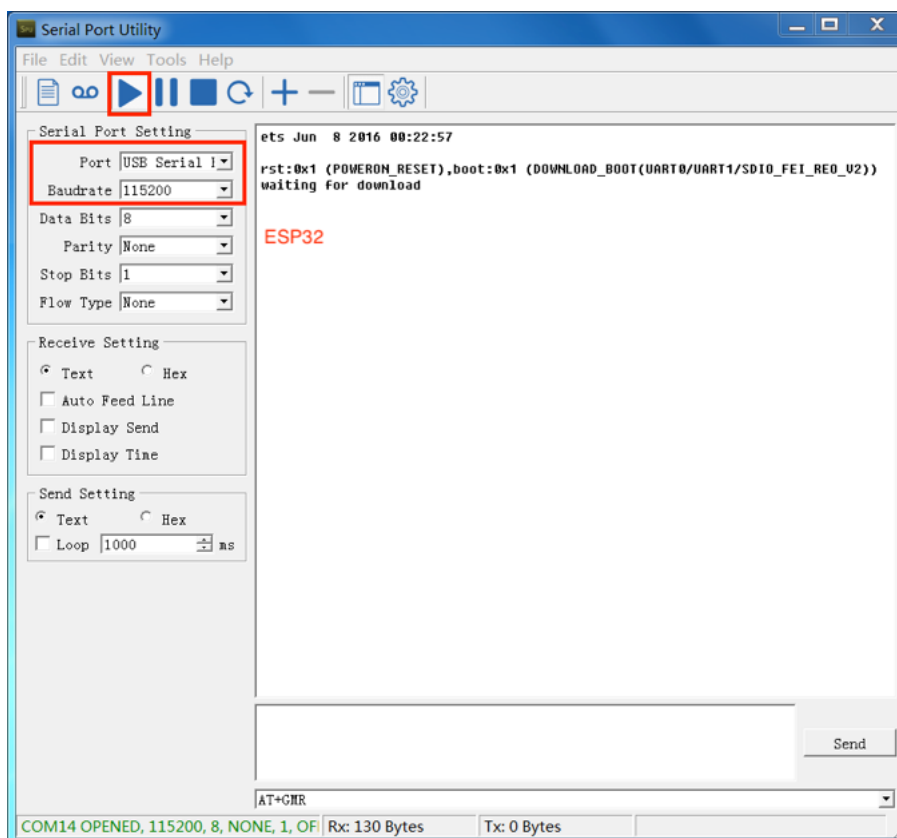


Fig. 100: Expected Result - ESP32 Series

- If the test result is as expected, which means the test was successful, users should save a screenshot of the test result.
- If the test result is not as expected, which means the test was not successful, users should perform the conductivity test again and ensure that the wiring is correct.

Test Report

Fixture manufacturers must perform the above-mentioned tests, and provide test reports or screenshots reflecting the results of these tests.

9.5 Appendix

Materials to Apply Fixtures

To accelerate the fixture manufacturing process, please provide the fixture manufacturers with the materials listed in the table below:

Table 13: Materials to apply fixtures

Material	Description
Module Gerber	Gerber files provide detailed information about the dimensions and positioning holes of the module.
Sample modules (for AT firmware downloading)	Sample modules can be useful for fixture manufacturers' testing. Please provide them on demand.
Serial Port Board	Please provide as many serial port boards as you actually need. For example, you should provide four serial port boards if you want to use a one-to-four fixture. (The serial board is ESP_Factory_Test_board V1.3.)
Wiring mode	Please inform the fixture manufactures whether you want to enable the Automatic Mode Switching or not. (By default, the Automatic Mode Switching on the Tool Side is not supported.)

Deliverable Items

The fixture manufacturers should deliver both of the items listed below:

Table 14: Deliverables

Deliverables	Description
Fixture Set	Fixture + serial ports + complete wiring. Notices: 1. If a one-to-four fixture is used, then there should be four serial boards in the bottom box, with complete wiring. 2. The serial board is ESP_Factory_Test_board V1.3.
Test Report	Test reports or screenshots reflecting the results of these tests.

Certification

Download certificates for Espressif products from [Certificates](#).

Warning: This document is not updated for ESP32C61 yet, so some of the content may not be correct.
 This warning was automatically inserted due to the source file being in the `add_warnings_pages` list.

10 Matter QR Code Generator

Matter QR Code Generator is used to generate QR codes that are used for provisioning Espressif's Matter devices. Integrated with BarTender, the generator enables label design and printing. With Matter QR Code Generator, you can configure label templates, select printers, and define data sources flexibly, meeting various QR code generation and printing needs in different scenarios. The generator also supports laser engraving machines over LAN for easier integration.

Download Link: [Matter QR Code Generator](#)

10.1 Software Directory

The directory structure of the Matter QR code generator is as follows:

- `bartender`: stores library dependency files
- `configure`: stores configuration files
- `data_output`: stores temporary output files
- `data_source`: stores files for local printing
- `files`: stores printing template files and scanning board firmware
- `esp_printer_main.exe`: the main executable file

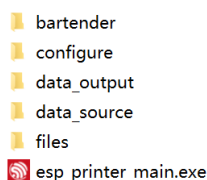


Fig. 101: QR code generator main interface (click to enlarge)

10.2 Get Started

Install BarTender

BarTender is the middleware of Espressif's Matter QR code Generator. Currently, Matter QR code Generator only supports [BarTender](#) 2022 and 2016 64-bit versions. During installation, make sure to choose the default path. You can select just the BarTender Designer module.



Fig. 102: Module selection (click to enlarge)

For more details about the installation process, refer to [Appendix II: BarTender \(2022\) Installation Process](#).

Edit Label Template

The label template defines the content and format of printed labels, and it can be edited using BarTender. The generator uses a default label template located in the the \\files\\matter directory. You may also customize this template to adjust the font, label size, or layout.

Note:

- Do not modify the label template filename.
- You can add or delete elements that are not bound to the data source, such as images, boxes, etc.
- Do not add or delete named data sources.

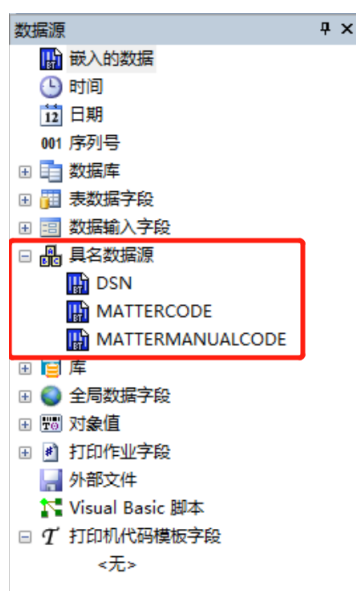


Fig. 103: Named data sources in the template (click to enlarge)

- The examples shown in the print interface are just static images. Your changes will not appear in the interface.

Tool Configuration

The configuration files are located in the directory `configure/config.conf`. You can open and edit them with Notepad.

Main Config	Subitem	Optional Value	Description
facConfig	rsiLimit	Recommended range: -30 ~ -80	The signal strength threshold that must be reached for surrounding products to be scanned before printing
	getMacType	[devboard, scan]	Provides two ways to enter device information: • devboard: Obtain MAC by receiving Bluetooth broadcasts via the scanning board • scan: Directly obtain device information using a barcode scanner
	print_enable	[0, 1]	Controls the printer's enabled status: • 0: Only retrieve information; printing is disabled • 1: Enables printing
SerialConfig	devPort	COM*	Scanning board serial port number
v2_scanboard (only for V2 scanning boards)	devBaud	115200	Scanning board baud rate
	scan_timeout	Default: 10	Scan timeout
	case_command	2	Fixed value
bartender	version	[2022, 2016]	BarTender Software Version ¹

10.3 Start Printing

¹ Currently, only version 2016 and 2022 are supported.

Interface

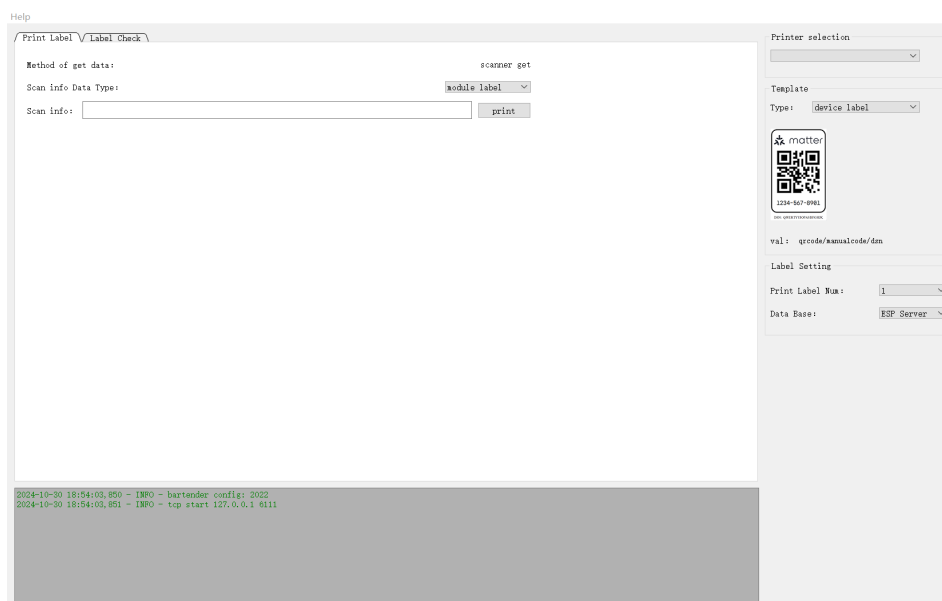


Fig. 104: Advanced Options (click to enlarge)

Interface Configuration

- **Printer selection:** The system printer is displayed by default. You can select a printer as needed
- **Template:** Choose the template file used for printing
- **Method of get data:** The method to get device information
 - **Scanner get:** Use a barcode scanner
 - * **Scan info Data Type:** The format of the content scanned by the barcode scanner
 - **Module label:** QR code on the Espressif module shield
 - **Device label:** The already printed device label
 - **MAC:** MAC address of Espressif products
 - **BLE Broadcast:** Use a scanning board
- **Print Label Num:** The number of labels to print. The maximum is currently 6
- **Data Base:** Data source
 - **ESP Server:** Retrieve QR code data from Espressif server
 - **Local excel:** Query data from a local table and copy it to data_source/matter_qrcode_data.xlsx in the required format:

说明: MAC列存放12个字符长度的MAC地址		
MAC	QR Payload	manual code
744DBDFBCC0C	MT:CWBA00QV173O303K400	32861103440

Fig. 105: Data storage format

- **Scanner data:** Retrieve information from scanned data (Currently only supported by Cyprus, as its device broadcast comes with MAC and QR code information).

Common Printing Methods

- Printing by scanning the shield QR code:

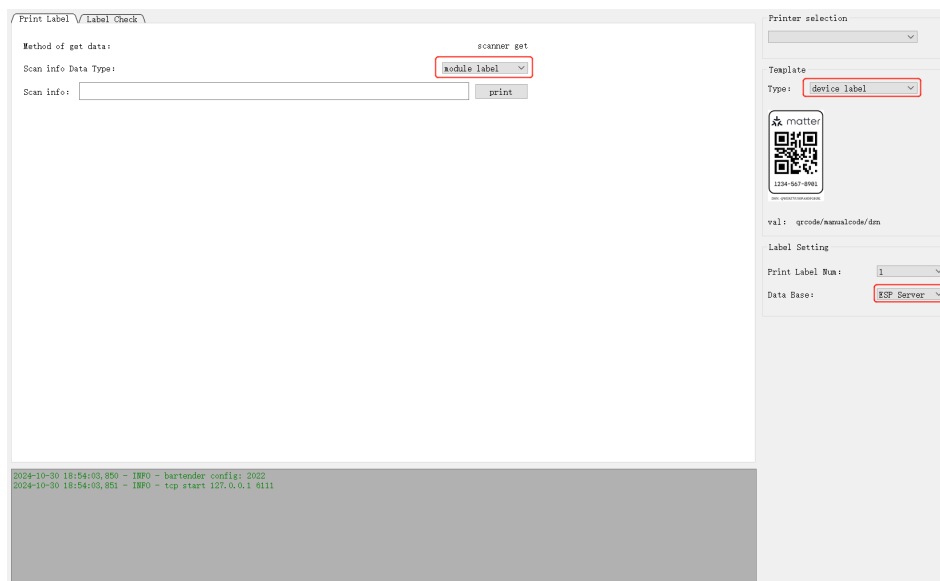


Fig. 106: Print by scanning the shield QR code (click to enlarge)

- Printing by scanning a printed label:

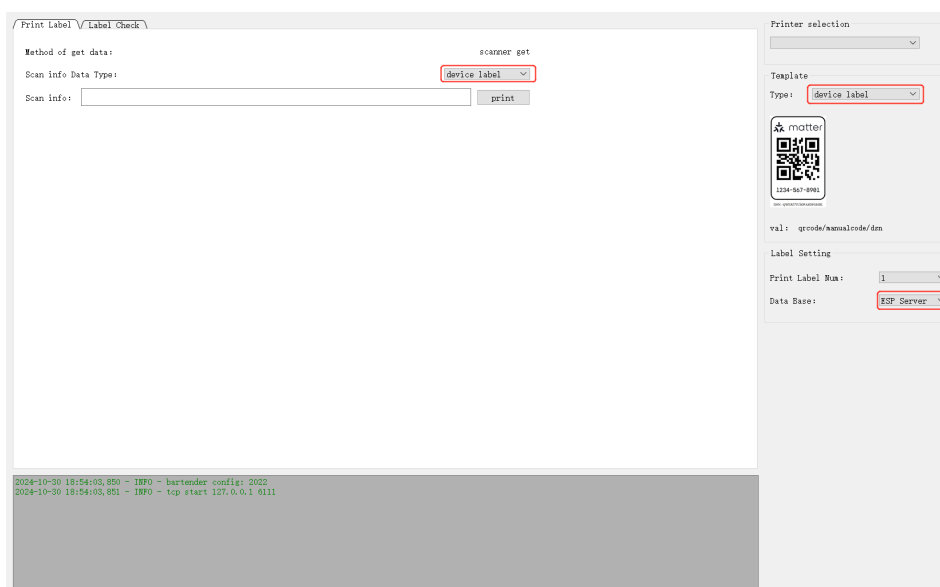


Fig. 107: Print by scanning a printed label (click to enlarge)

10.4 Check Printed Labels

The purpose of checking printed labels is to ensure that the device information matches the printed QR code. To do this, a scanning board is required to scan the device's Bluetooth broadcast signal.

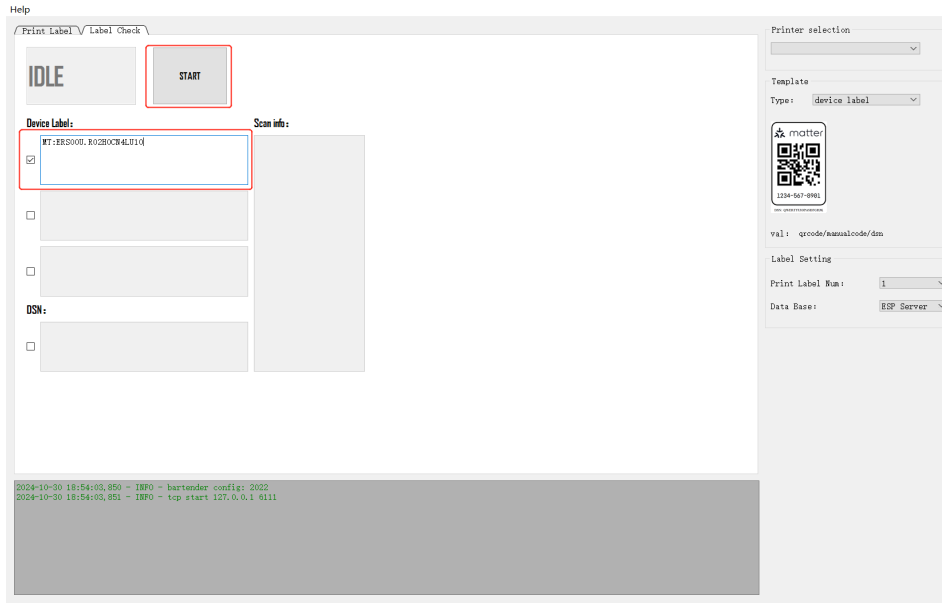


Fig. 108: QR code check (click to enlarge)

- To check the QR code, you need to use the configuration method of the scanning board, which corresponds to the Method of `get data: BLE boardcase` in the `Print Label` interface.
 - In the configuration file `facConfig`, set `getMacType = devboard`.
- Check the checkbox based on the number of device codes to be checked, so as to enable the corresponding number of device labels.
- To perform a DSN check (only applicable for Cyprus), check the checkbox to enable this feature.

10.5 Integrate Laser Marking

Currently, the QR code can be obtained over LAN, enabling integration with the laser marking machine.

Configuration

Table 15: TCPserverConfig

Configuration Item	Configuration Value	Description
server_enable	1	Enable LAN for QR code retrieval
ip	127.0.0.1	LAN address. If the laser marking host and this host are on the same PC, the loopback address can be used
port	6000	TCP communication port
qr_req_string	get_qrcode	The command to request QR code. This can be adjusted based on the configuration of the laser marking machine
manual_req_string	get_manualcode	The command to request manual code. This can be adjusted based on the configuration of the laser marking machine
dsn_req_string	get_dsncode	The command to request dsn code. This can be adjusted based on the configuration of the laser marking machine

10.6 Appendix I: Flash Scan Board Firmware

- To flash the scanning board firmware, the ESP32-C3 series development boards are required. Select the development board based on your needs.

- bin file path: ./files
- Flash address: 0x0

Flash tool download: [Click here to download the flash tool](#)

10.7 Appendix II: BarTender (2022) Installation Process

The following figures show the whole process of BarTender installation (Take BarTender 2022 as an example):

1. Check the advanced installation option checkbox.



Fig. 109: Choose advanced installation options (click to enlarge)

2. Use the default installation path.

高级安装选项

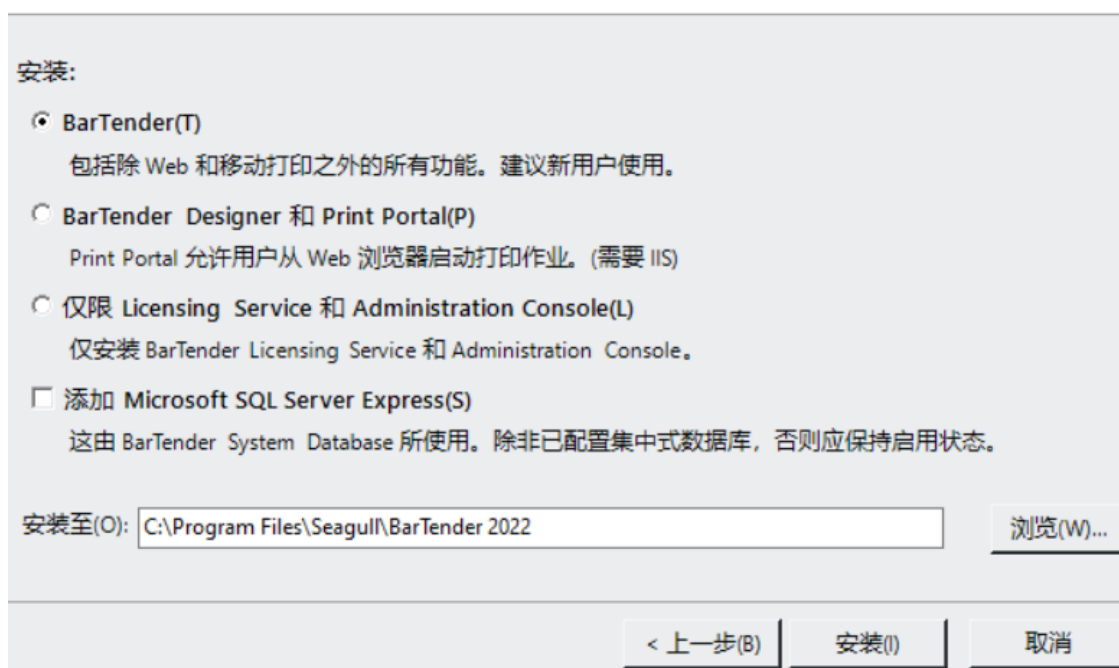


Fig. 110: Default installation path (click to enlarge)

3. The installation process is as follows.



Fig. 111: BarTender Installing (click to enlarge)

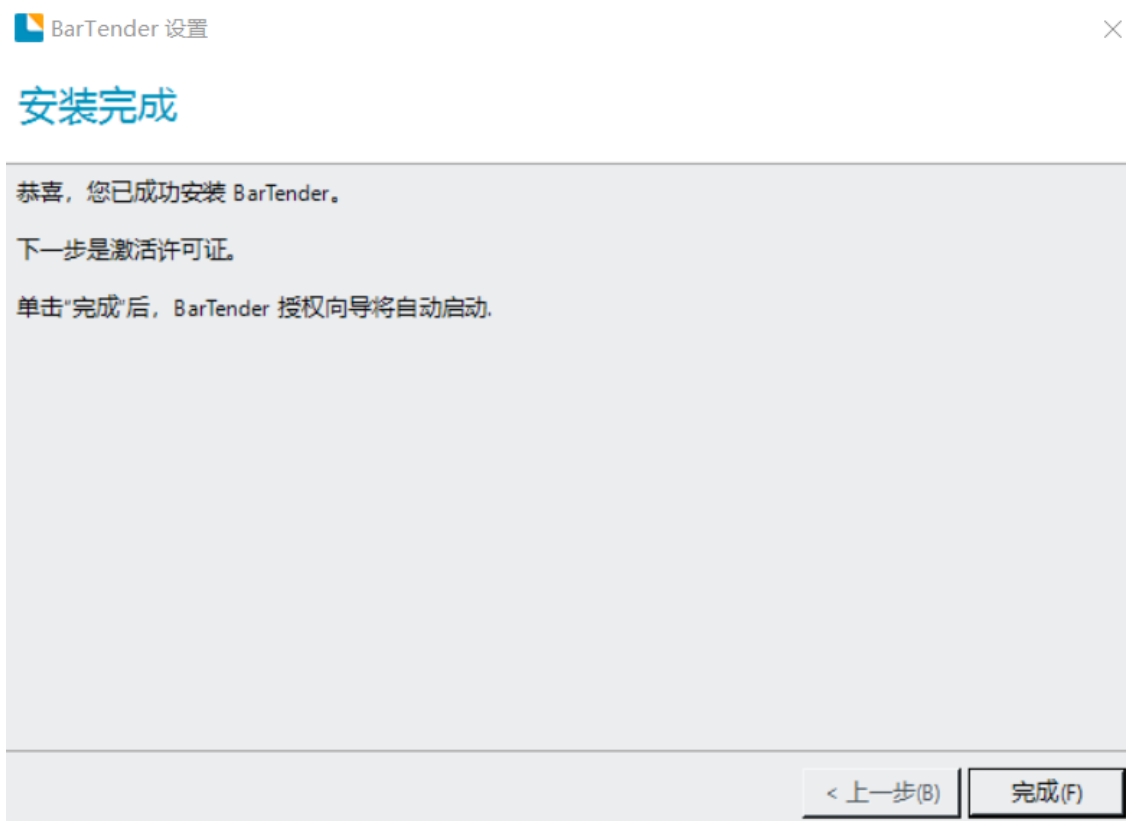


Fig. 112: Installation completed (click to enlarge)

4. Enter the serial number to activate BarTender.



Fig. 113: Enter serial number (click to enlarge)

11 FAQ

- *RF Testing FAQs* offers answers to common questions about *EspRFTTestTool Toolkit* and *RF Test Items*.
- *Flash Download Tool FAQs* cover common questions about the *Flash Download Tool User Guide*.
- *Espressif Production Testing Guide FAQs* provide answers to common questions about the *Espressif Production Testing Guide*.

11.1 RF Testing

1. What should I do if the EspRFTTestTool Toolkit fails to flash?

The chip might not have entered download mode correctly. Follow these steps to troubleshoot:

- Check the log: Use a serial tool (such as `sscom`, [Serial Port Utility](#)), select the correct baud rate, and check the log after powering up the chip.
- Confirm download mode: When the chip enters download mode, it typically displays “wait for download.”
- Check the connections: If no log is printed, ensure that the power supply and UART connections are functioning properly.

2. How can I confirm whether the firmware was successfully flashed?

Even if the flashing tool indicates success, the firmware might not have been flashed correctly. Follow these steps to verify:

- Check the log: Close the serial port used by the flashing tool, open a serial tool (such as `sscom`, [Serial Port Utility](#)), select the correct baud rate, and check the log.
- Enter working mode: Pull up the Boot pin and re-power the chip to enter working mode.
- Confirm flash success: Check if the log shows continuous reboots or matches the expected behavior based on the firmware documentation to confirm if the flashing was successful.

3. What should I do if the running traffic fails in the Wi-Fi Adaptivity Test?

If running traffic fails, consider the following possible causes and solutions:

- Firmware issues: Ensure that the firmware was flashed successfully.
- Network issues: Check whether the router (AP) network is stable and connections are smooth.
- Connection delays: If the connection is slow, wait a few seconds and restart running traffic.
- Serial testing: If the issue persists, consider testing via serial commands.

11.2 WFA Certification Test

1. How can I get the USB port name of the device?

Run the command `ls /dev/ttyUSB*` in the terminal to see the USB port name.

2. How can I get the MAC address of the DUT?

- Open `minicom` with the command `minicom -D /dev/ttyUSB*`;
- Type `query` and the MAC address of the DUT will be shown as `dut_mac`.

3. How do I flash the enterprise certificate?

The certificate is already included in the firmware, so you do not need to flash it separately.

4. Why isn't the tool starting?

Check the Python version and ensure the toolchain is fully installed.

5. Why is the tool script not detecting UCC commands after starting?

Ensure that the IP address is correctly configured on your computer.

6. What should I do if the DUT shows garbled output and is unresponsive to read/write commands?

Confirm that the DUT is flashed with the correct bin files and check that the power supply is working properly.

11.3 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:

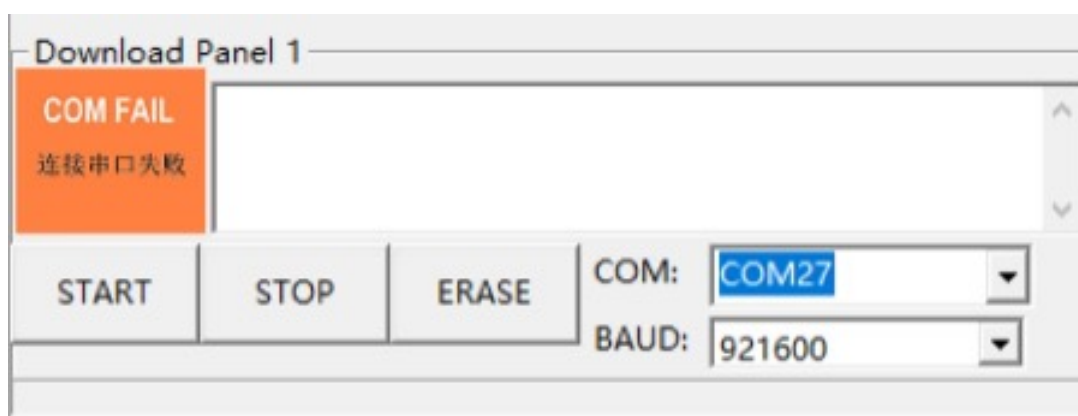


Fig. 114: Connection Failure of Serial Port

- 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?

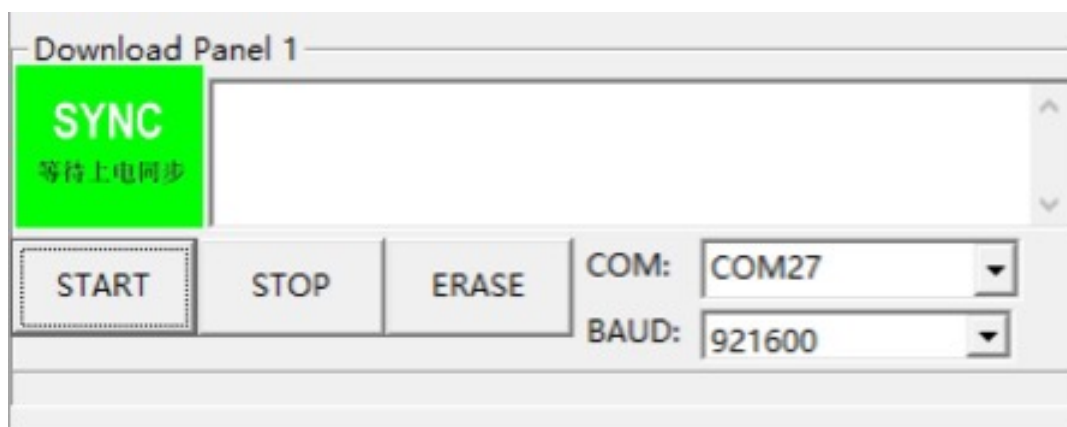


Fig. 115: Download Panel

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. 116: 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.

11.4 Espressif Production Testing Guide

1. Why it is necessary to set up an evaluating environment?

To ensure smooth mass production testing, the test environment must be evaluated beforehand. This is to confirm several aspects: stable power supply (including power to the DUT and the signal board), that the signal board and production test baseboard meet requirements, and to eliminate potential interference from the surrounding environment.

2. What should be done if RX FAIL occurs after testing and `fb_rssi` and `dut_rssi` are outside the normal range?

If RX FAIL occurs after testing and `fb_rssi` and `dut_rssi` are greater than 60 or less than -30, the following measures can be taken: increase the distance between the signal board and the module under test, or add a 30 dB attenuator on the signal board side.

3. How often does the signal board need to be calibrated? How can interference between signal boards be avoided?

The MAC address and production date of the board are given at the back of the signal board. Note that the signal board must be recalibrated every year, because the long operating time of components, such

as crystal oscillators, may lead to measurement deviations. Only ONE signal board must be used in an independent environment or RF-shielded environment to avoid interference.

12 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](#)

13 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.