ESP32-H2

User Guide of ESP Test Tools

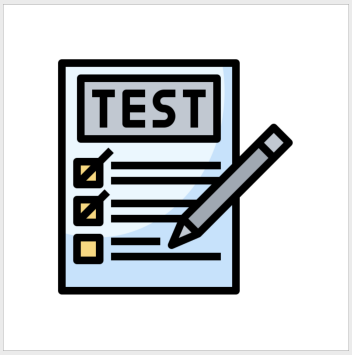
ESPRESSIF

Table of contents

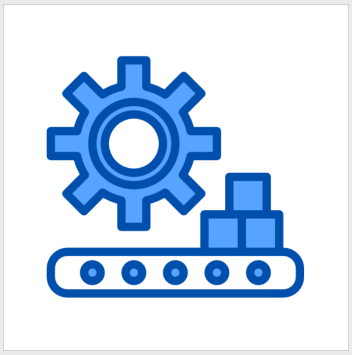
This repository provides comprehensive resources to support the development and production of products based on Espressif chips and modules.

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.

RF Test Guide



RF Test Items



Production-stage Tools

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

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.

802.15.4 Test

- *802.15.4 Non-Signaling Test* directly controls the device to transmit specific signals without requiring a network connection. It evaluates key performance metrics such as transmit power, spectrum characteristics, and error rate, ensuring reliable communication quality in IoT applications.

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
Bluetooth LE Non-Signaling Test	Y	Y	Y
Bluetooth LE DTM Test	Y	–	–
Bluetooth LE Adaptivity Test	Y	–	–
Bluetooth LE Blocking Test	Y	–	–
802.15.4 Non-Signaling Test	Y	Y	Y

2 EspRFTestTool Toolkit

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

- *EspRFTestTool*: 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: `EspRFTestTool toolkit`

The zip file not only includes the EspRFTestTool 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 **EspRFTestTool toolkit** refers to the collection of the three tools, while the **EspRFTestTool** refers to this single tool.

2.1 EspRFTestTool

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

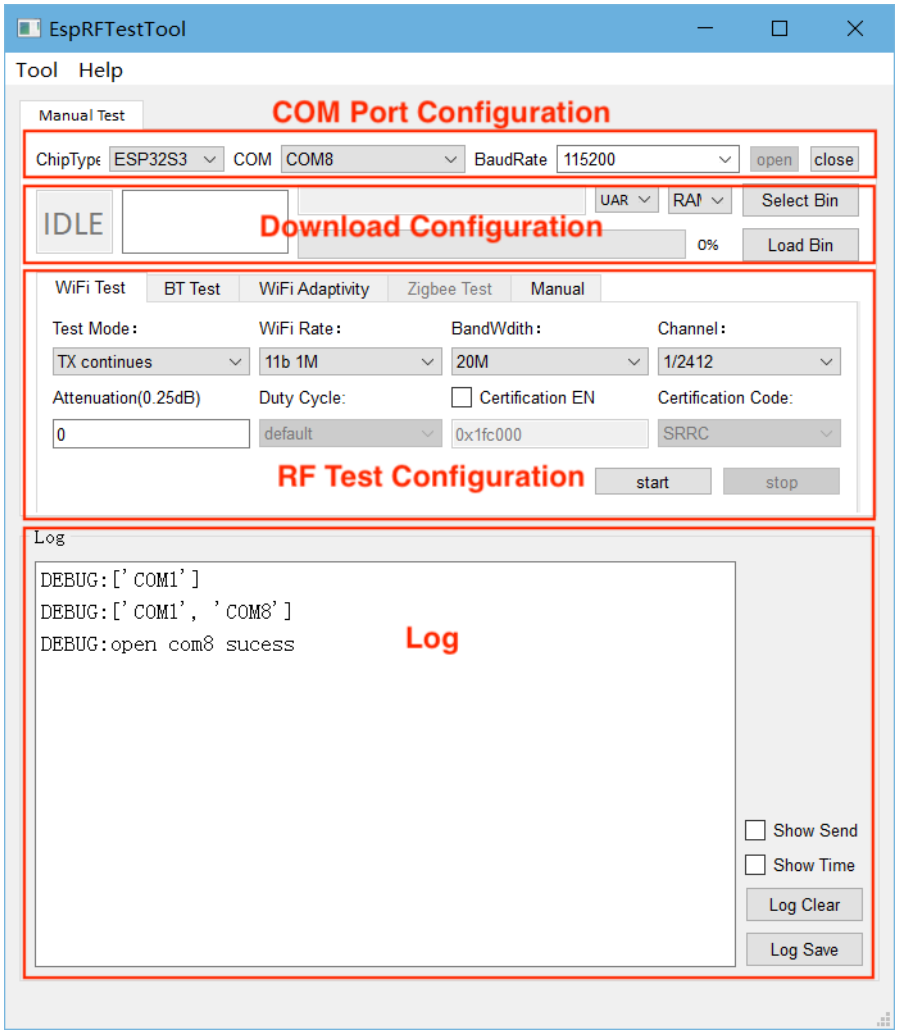


Fig. 1: EspRFTestTool

COM Port Configuration Area

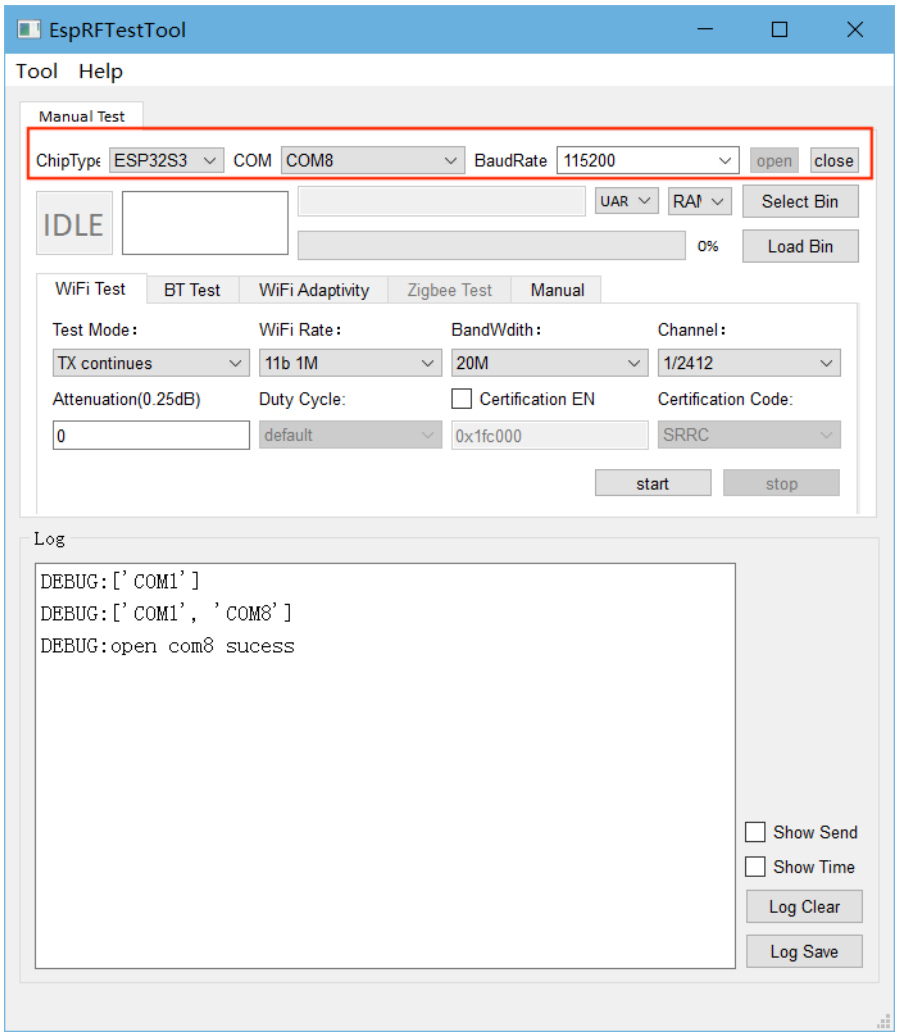


Fig. 2: EspRFTestTool COM Port Configuration Area

- **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

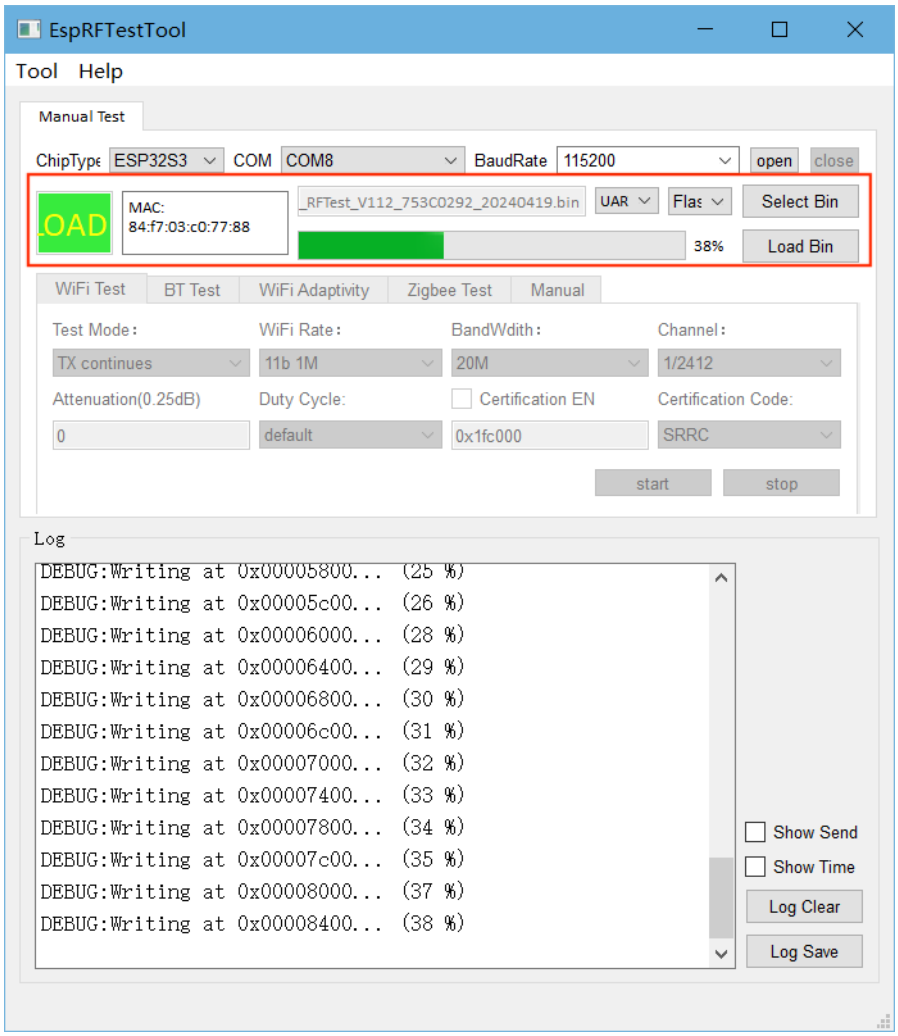


Fig. 3: EspRFTestTool Download Configuration Area

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, EspRFTestTool 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

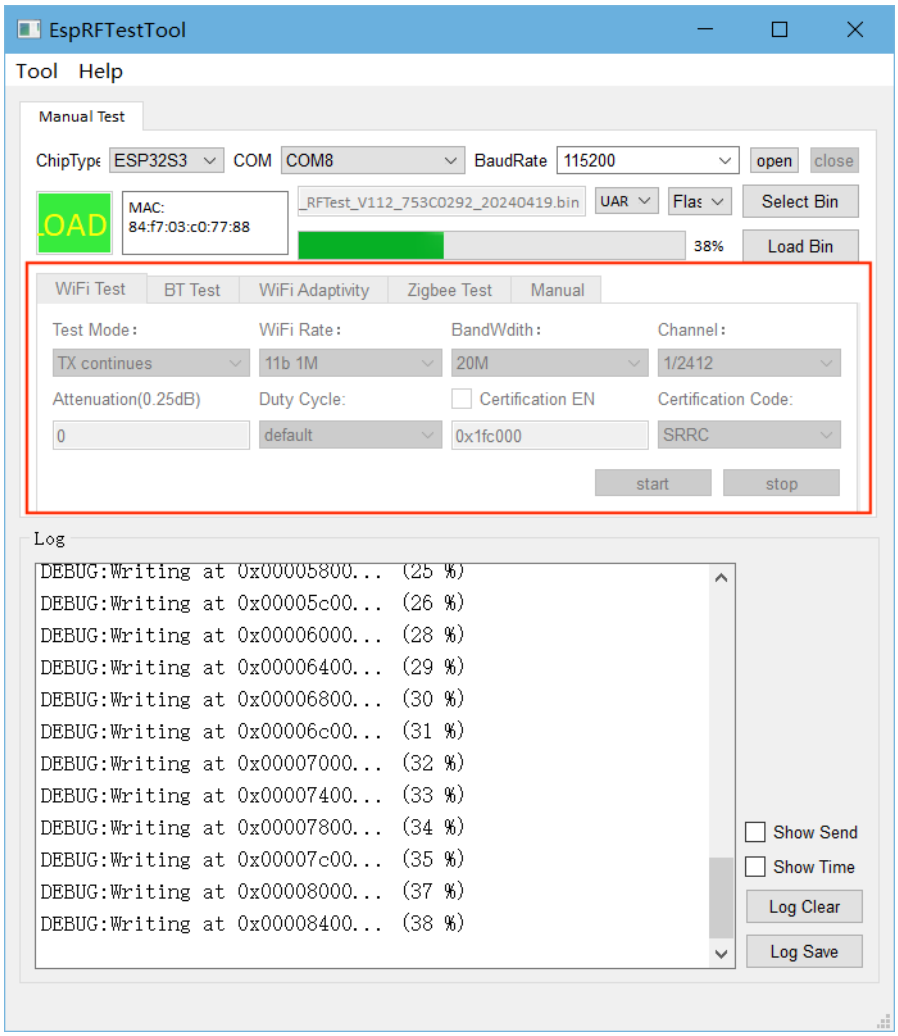


Fig. 4: EspRFTestTool RF Test Configuration Area

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.

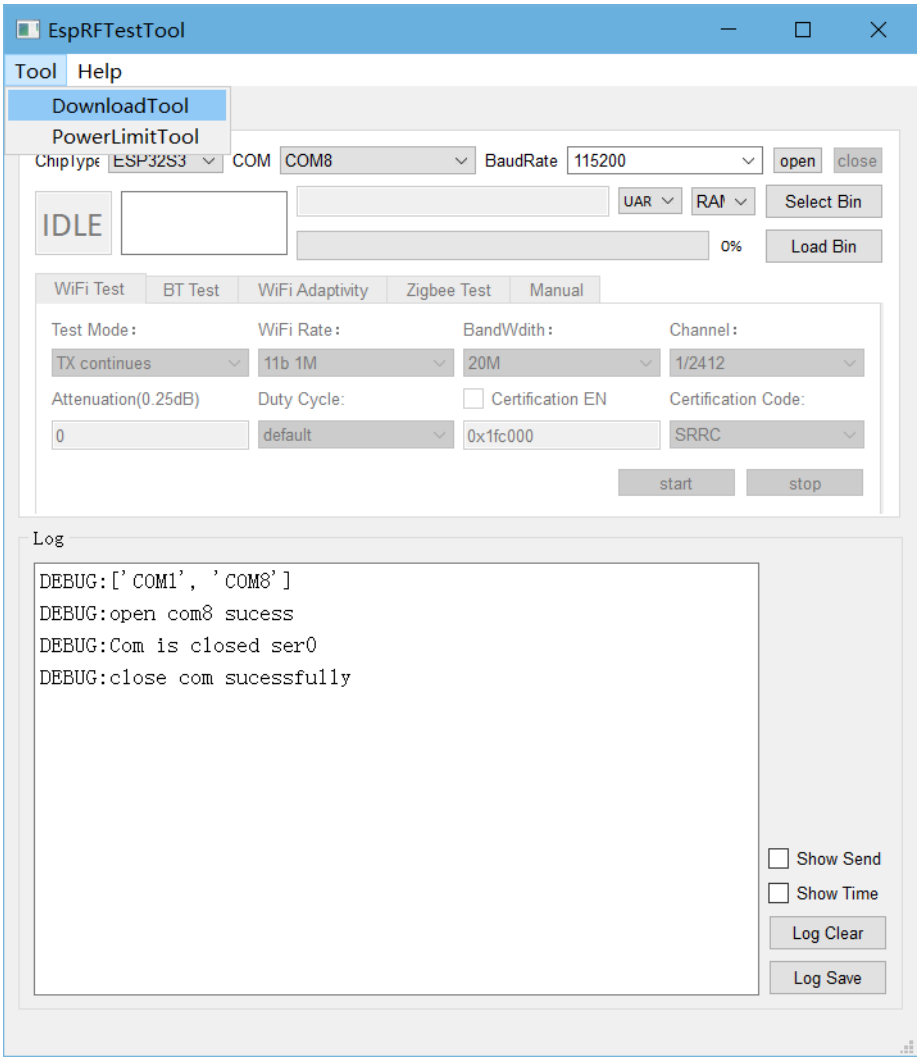


Fig. 5: Entry to DownloadTool

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.

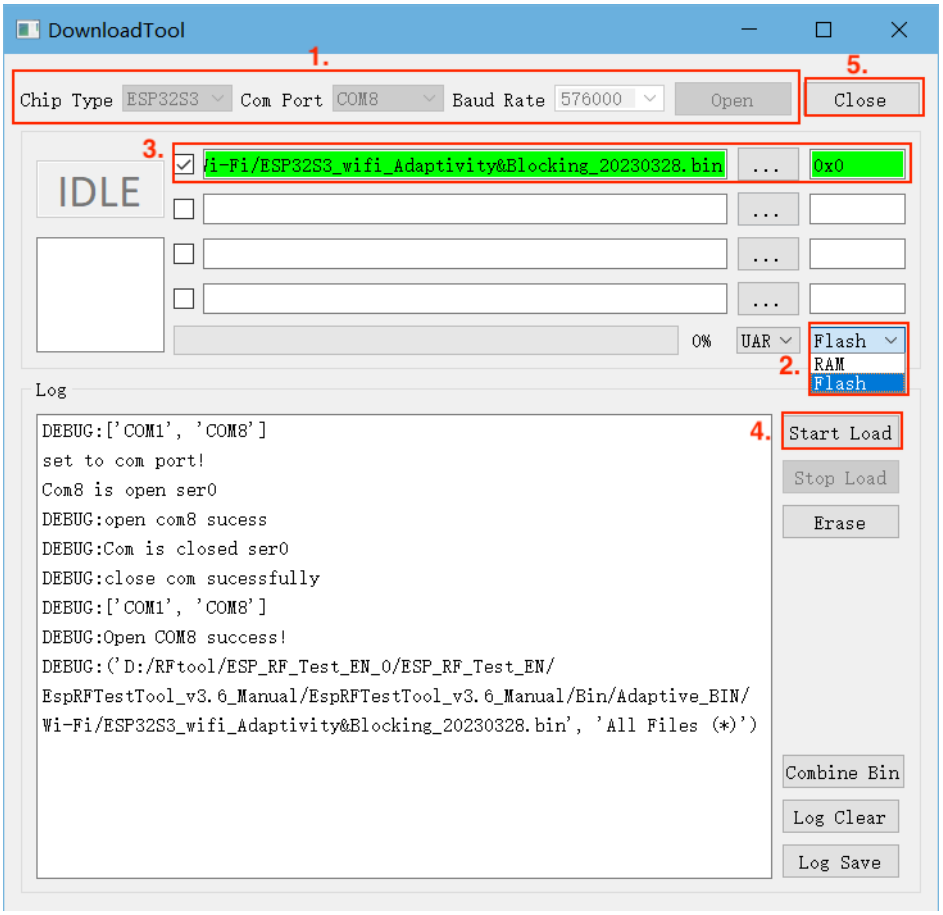


Fig. 6: DownloadTool Interface

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 EspRFTestTool, click `Tool`, and select `PowerLimitTool` from the dropdown box to open PowerLimitTool.

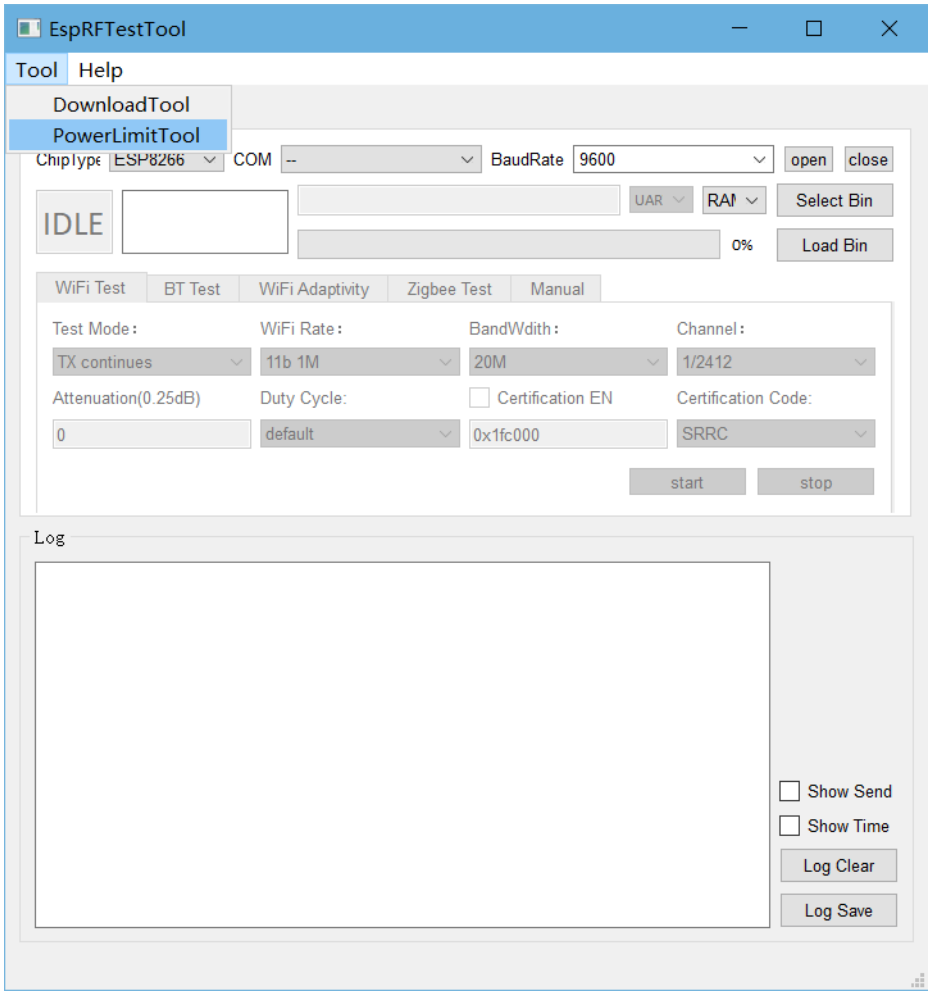


Fig. 7: Entry to PowerLimitTool

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).

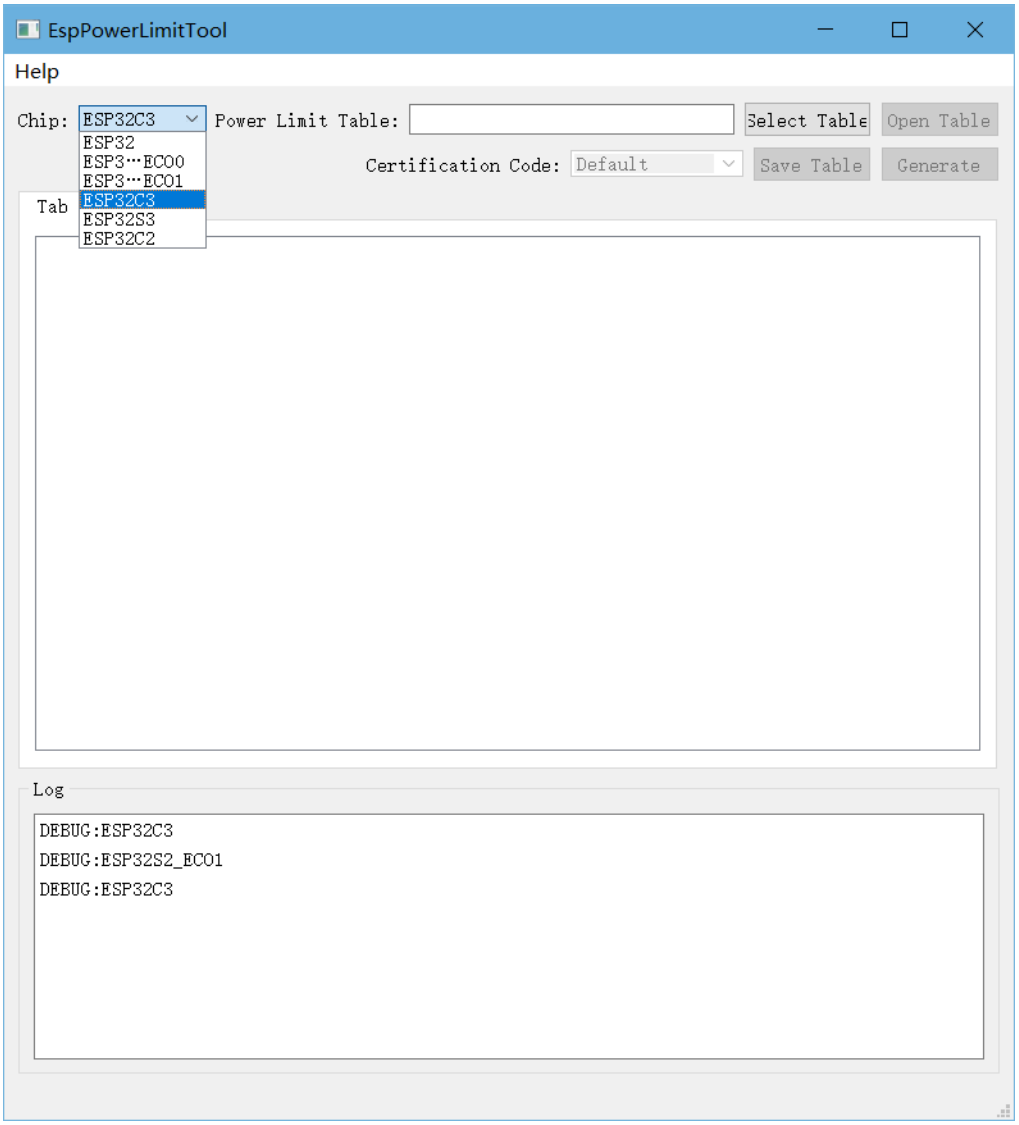


Fig. 8: PowerLimitTool Main Interface

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

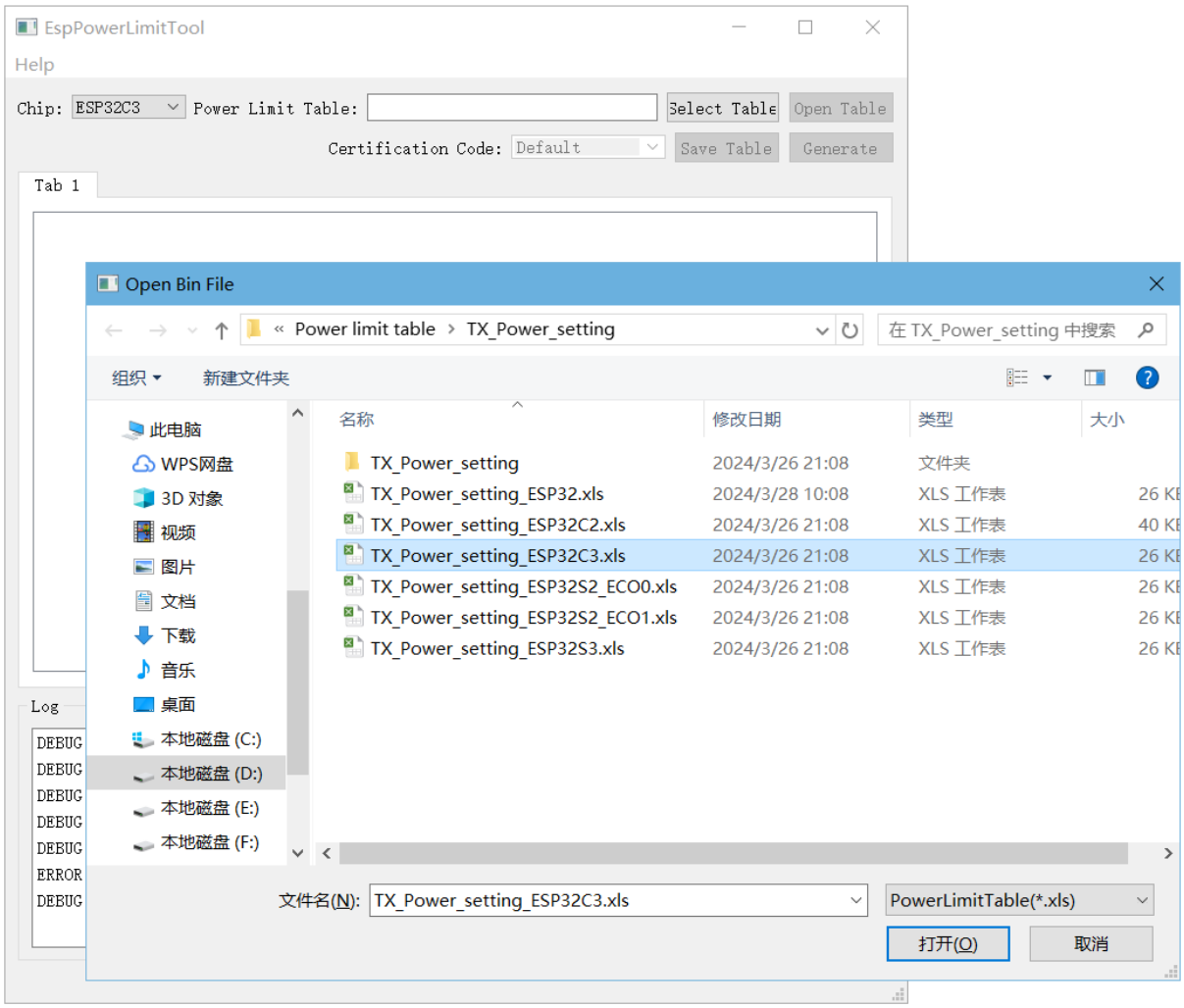


Fig. 9: Importing TX Power Setting Table

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.

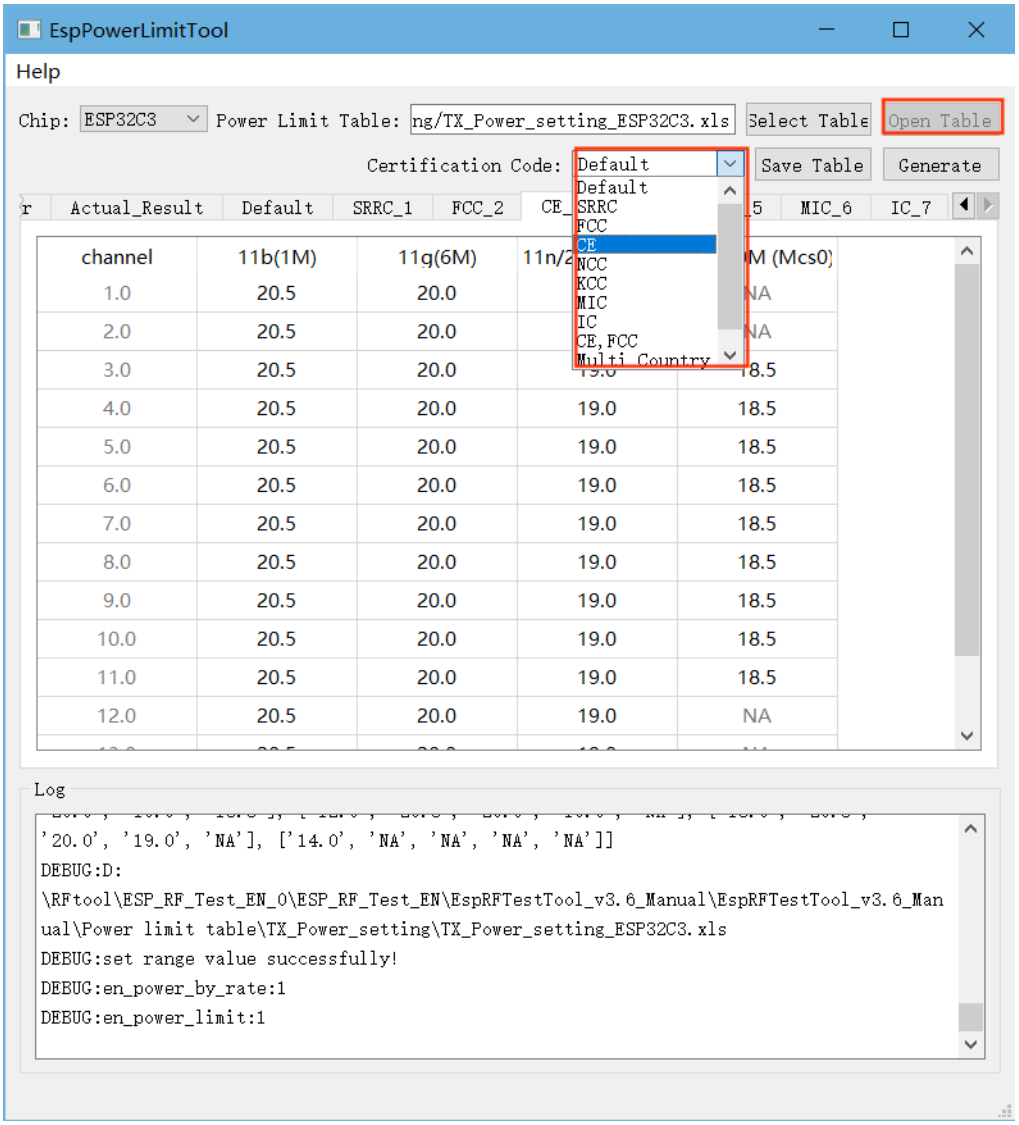


Fig. 10: Modifying TX_Power_Setting

Note: Description of TX Power Setting Table parameters:

1. **Config_Switch**: Enable `Power_By_Rate` and `Power_Limit`. Both are set to `Yes` by default, indicating they can be adjusted.
2. **PowerByRate_TargetPower**: Target power for each rate. It is recommended to keep the default value.
3. **Country_Table**: Currently supported countries (regions). It is extensible.
4. **Actual_Result**: Actual power of the module. The target power is used by default.
5. **Default**: Power configuration in the "Default" country code, usually used to identify the power configuration before setting the country code.
6. **SRRC_1**: Power configuration of the "SRRC" country code, applicable to Mainland China.
7. **FCC_2**: Power configuration of the "FCC" country code, applicable to the United States.
8. **CE_3**: Power configuration of the "CE" country code, applicable to Europe.
9. **NCC_4**: Power configuration of the "NCC" country code, applicable to Taiwan.
10. **KCC_5**: Power configuration of the "KCC" country code, applicable to South Korea.
11. **MIC_6**: Power configuration of the "MIC" country code, applicable to Japan.
12. **IC_7**: Power configuration of the "IC" country code, applicable to Canada.

Note: How to modify power values:

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.

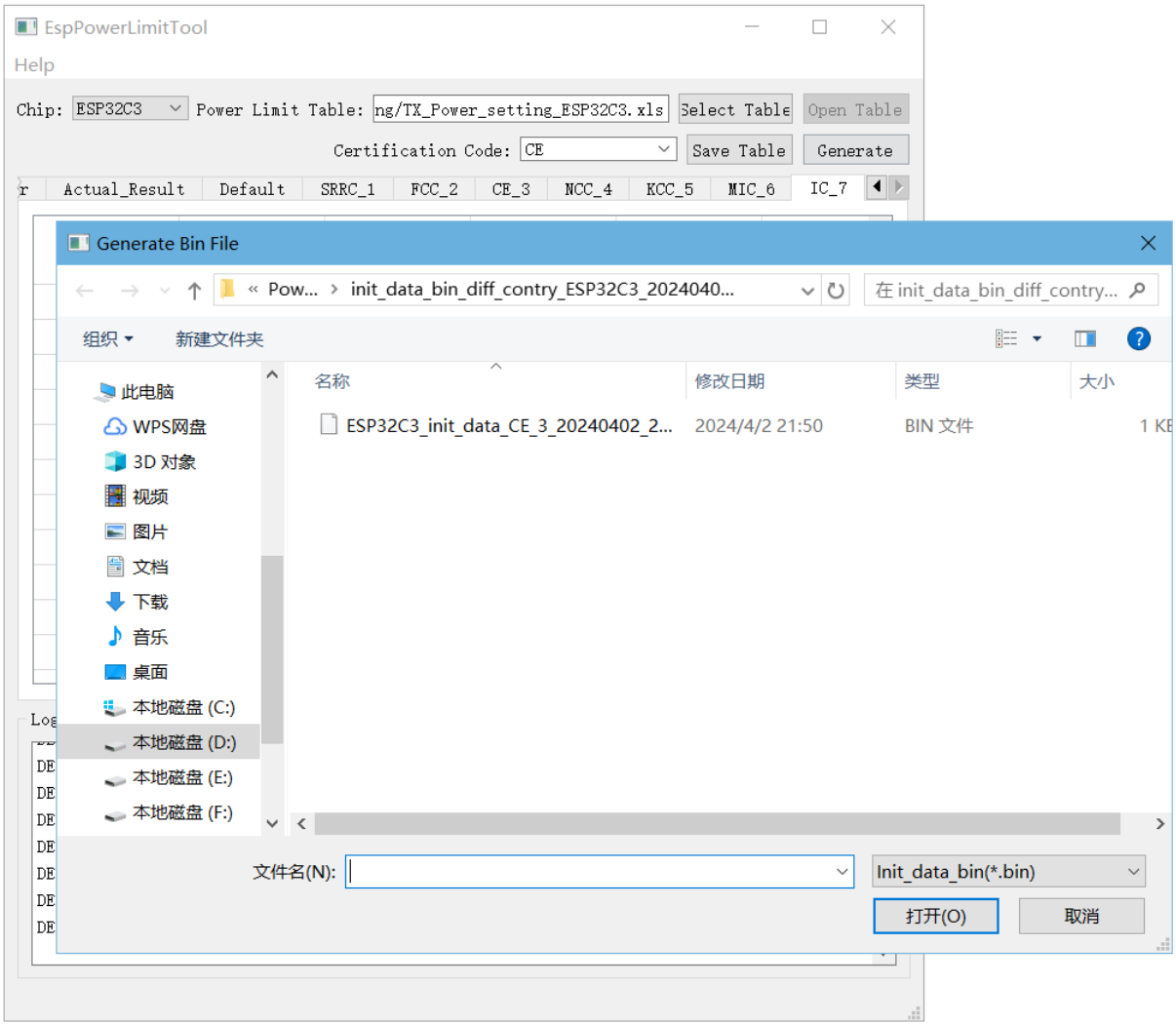


Fig. 11: Generate phy_init_bin File

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 bing 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 ESP32-H2 RF Non-Signaling Test 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*.

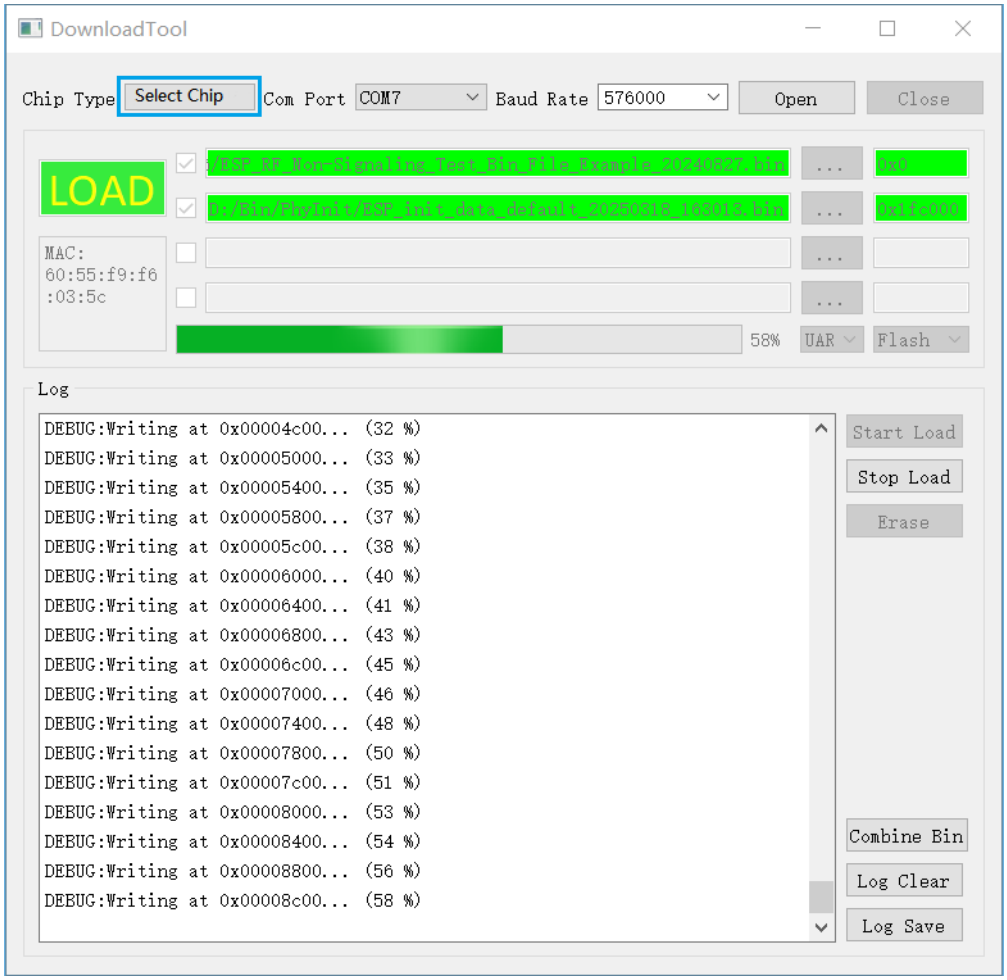


Fig. 12: Flash phy_init_bin File

6. Use a Wi-Fi tester to measure the output power and check whether phy_init_bin is effective.

- Open *EspRFTestTool*.
- 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`.

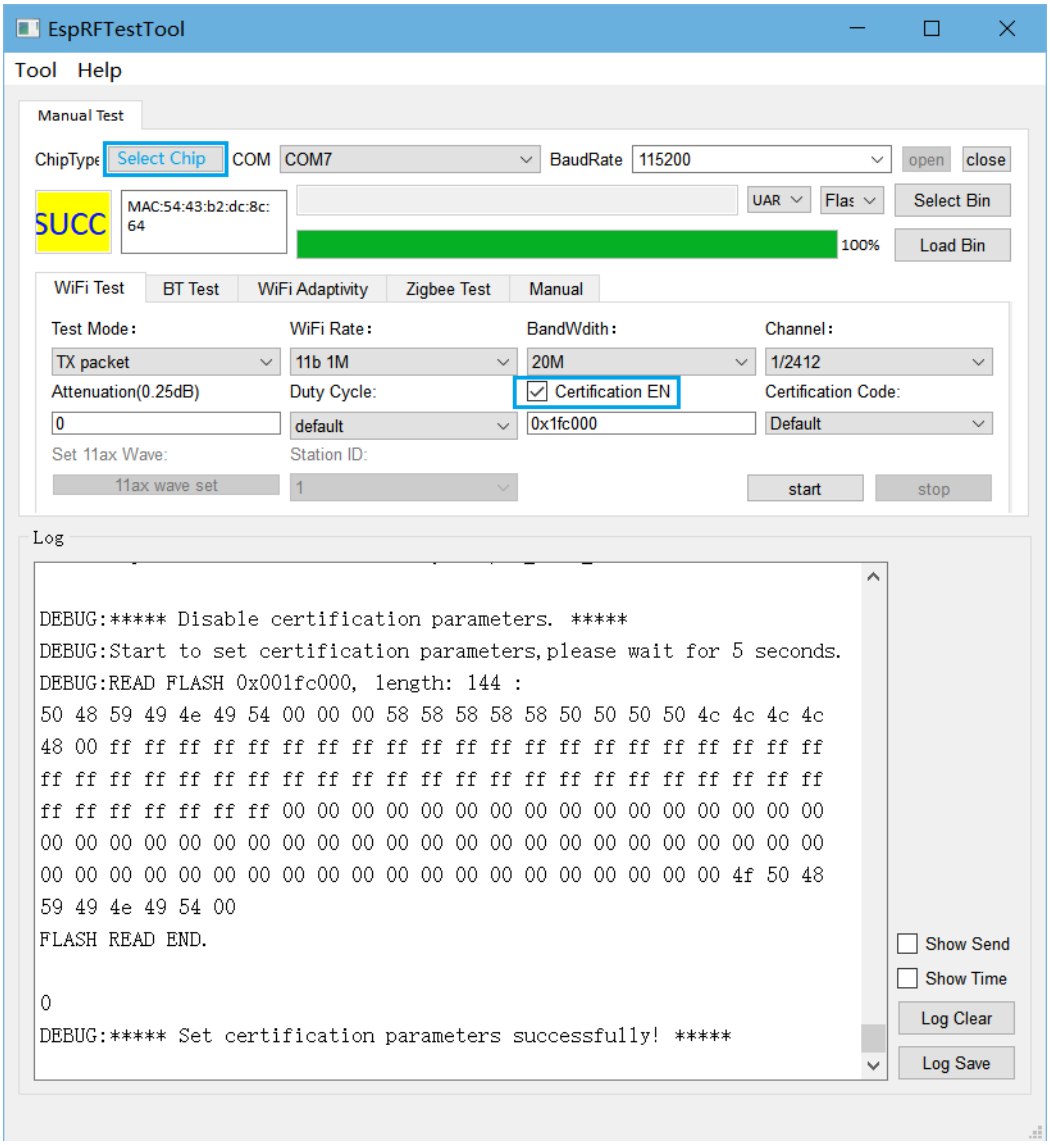


Fig. 13: RF Test Configuration

3 RF Test Items

3.1 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.

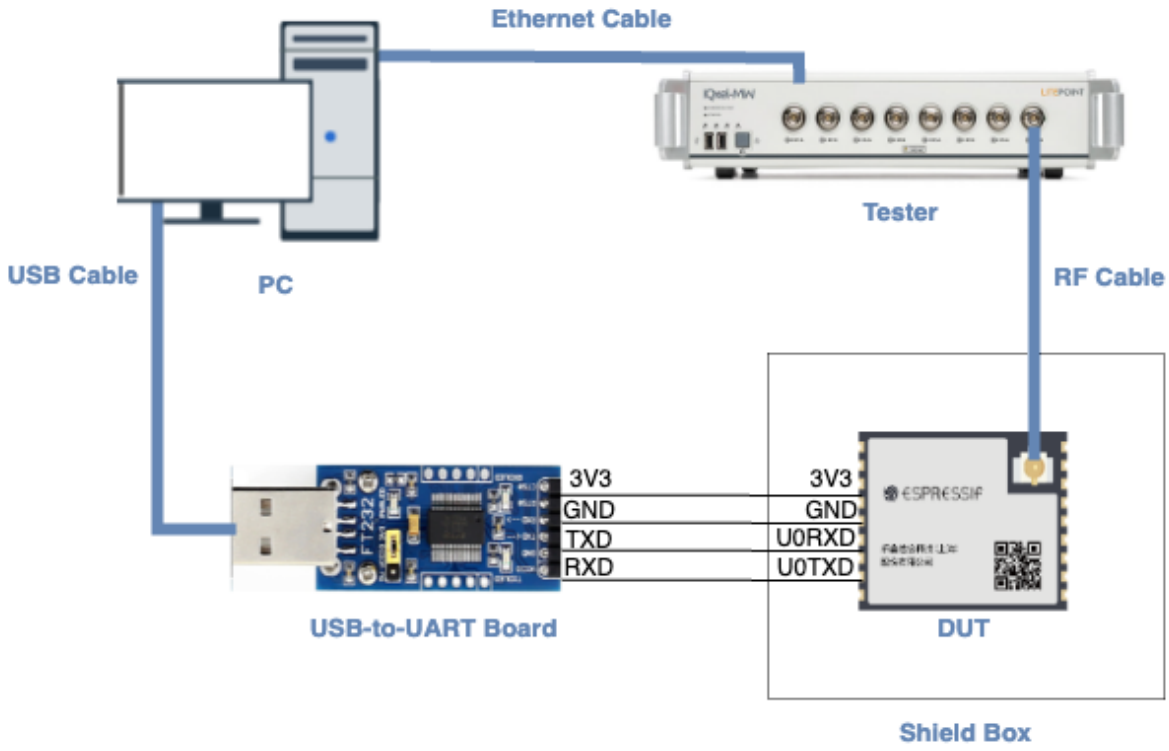


Fig. 14: Test Environment Setup

- **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 EspRFTestTool 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-H2 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-H2 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.

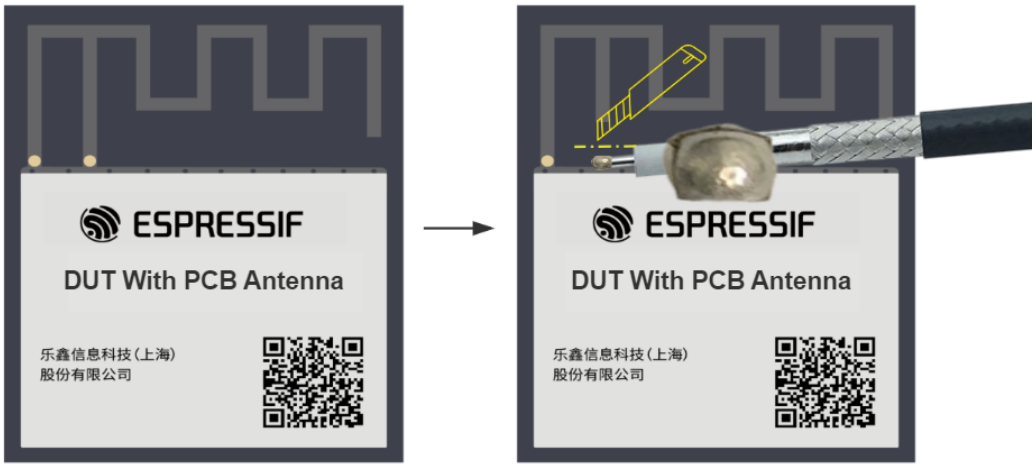


Fig. 15: Soldering RF Connection Cable to Module with Onboard PCB Antenna

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 ESP32-H2 RF Non-Signaling Test Firmware to `Flash` via `UART`.

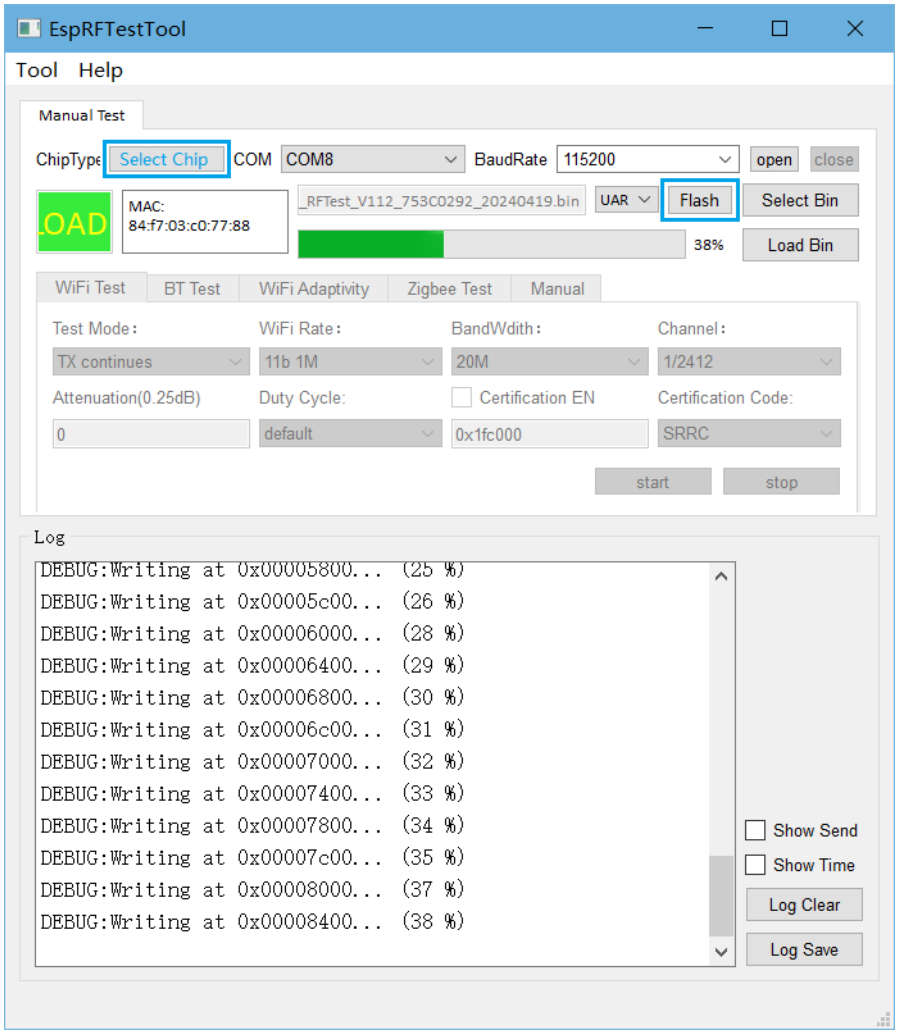


Fig. 16: ESPRFTestTool Configuration

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-H2 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.

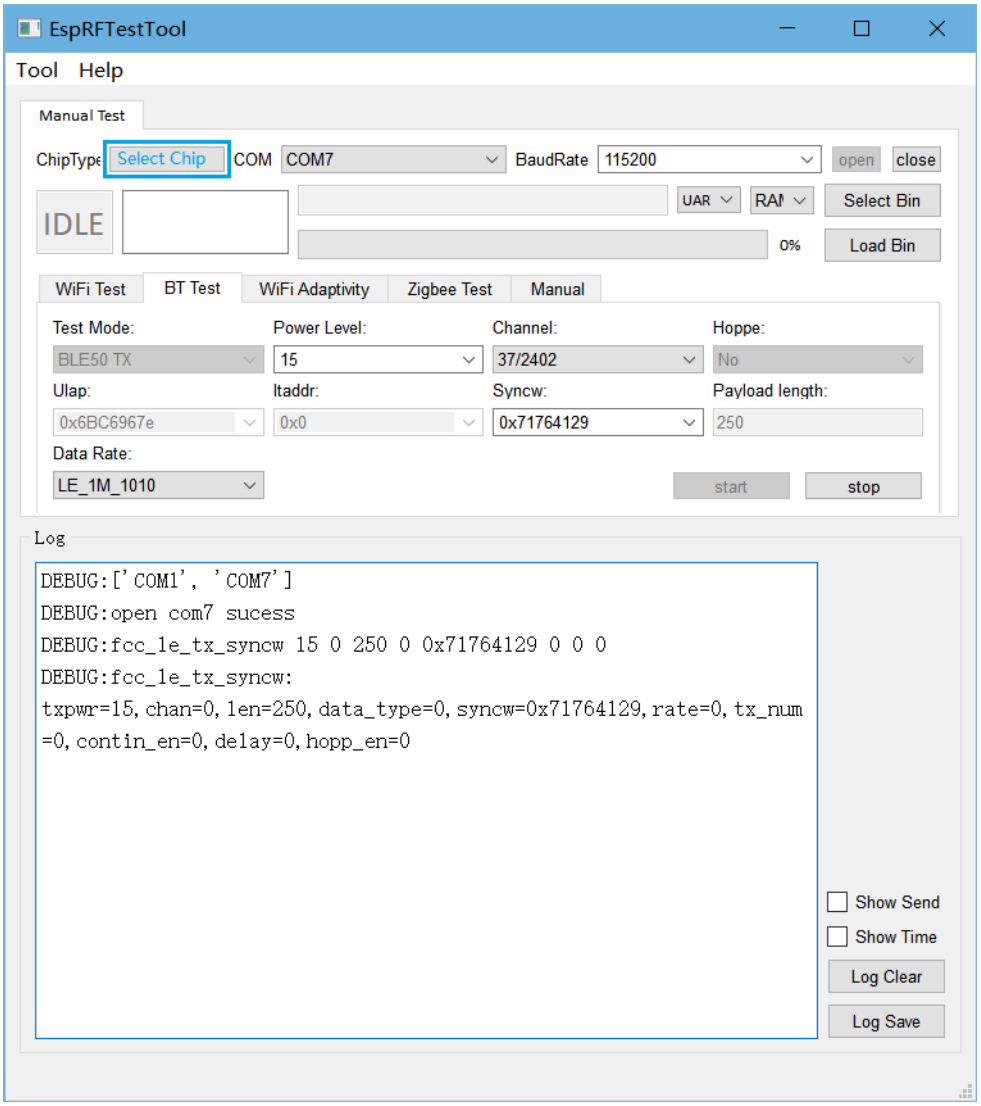


Fig. 17: Bluetooth LE TX Performance Test

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-(Res[1]/Sent_Packet_Numbers)]*100%<=30.8%
- RSSI of each package = Res[11]/(Res[1])

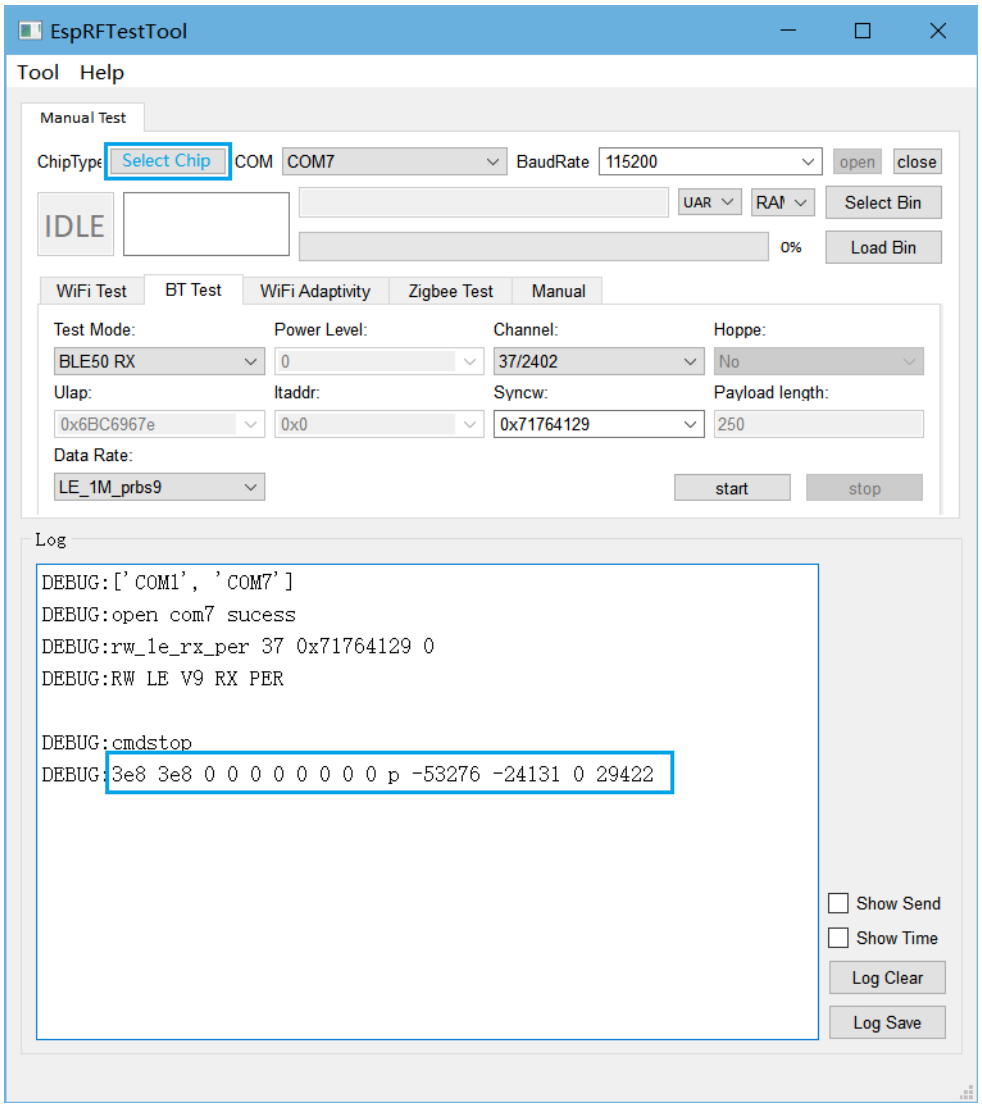


Fig. 18: Bluetooth LE RX Performance Test

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-H2, serving as a reference for RF debugging or testing.

Table 2: ESP32-H2 Bluetooth LE TX Power Level

Power Level	Bluetooth LE TX Power (dBm)
0	-24
1	-21
2	-18
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 EspRFTestTool toolkit uses the channel index to identify channels.

Table 3: ESP32-H2 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

3.2 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

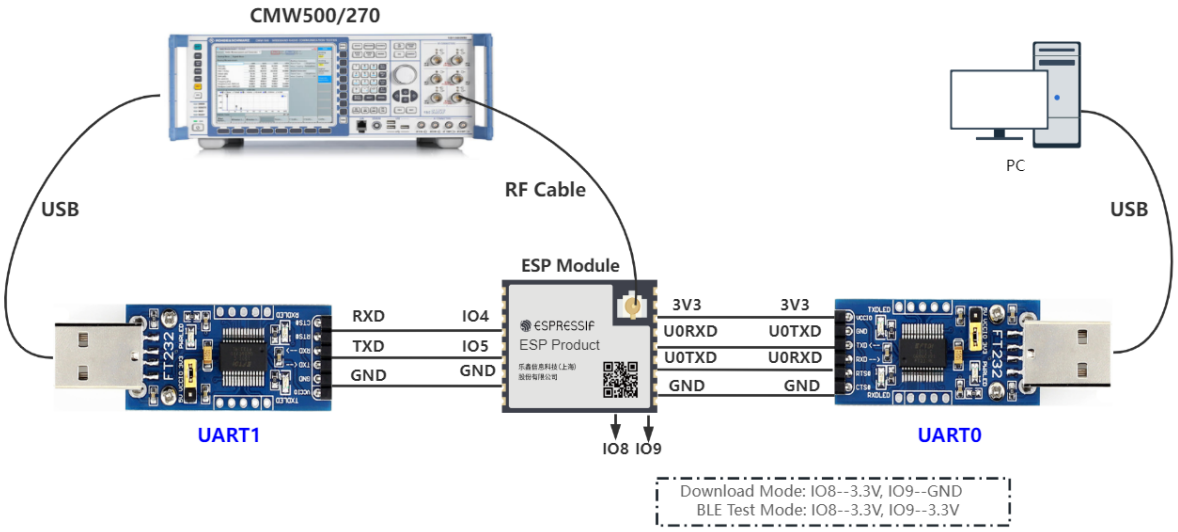


Fig. 19: Test Environment Setup

- **PC** is connected to the USB-to-UART board via USB. The PC needs to have the EspRFTestTool 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-H2 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-H2 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.

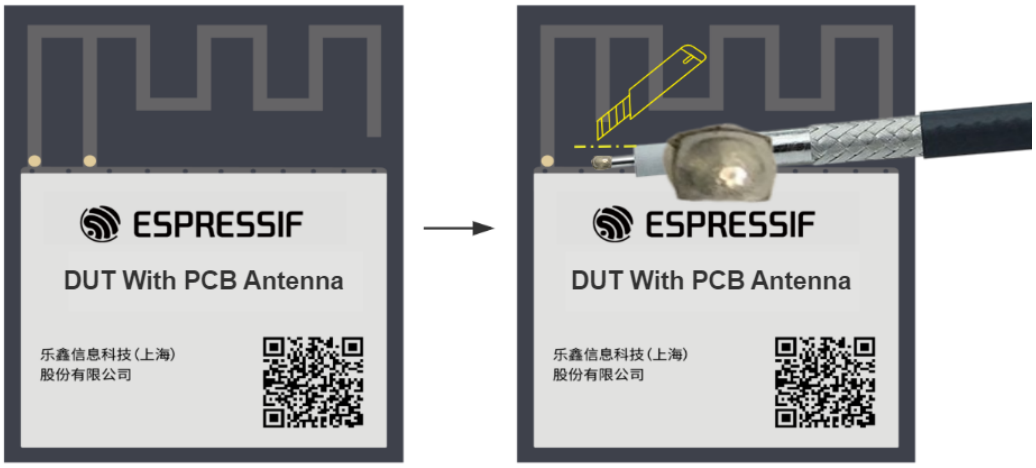


Fig. 20: Soldering RF Connection Cable to Module with Onboard PCB Antenna

Flash Firmware

1. Open *DownloadTool*.
2. Set `ChipType`, `Com Port`, `Baud Rate`, click `Open`, and select to download to `Flash`.
3. Flash the ESP32-H2 Bluetooth LE DTM Test Firmware bin file to the following address via `UART`.

Bin File	Flash Address
ESP32-H2 Bluetooth LE DTM Test Firmware	0x0

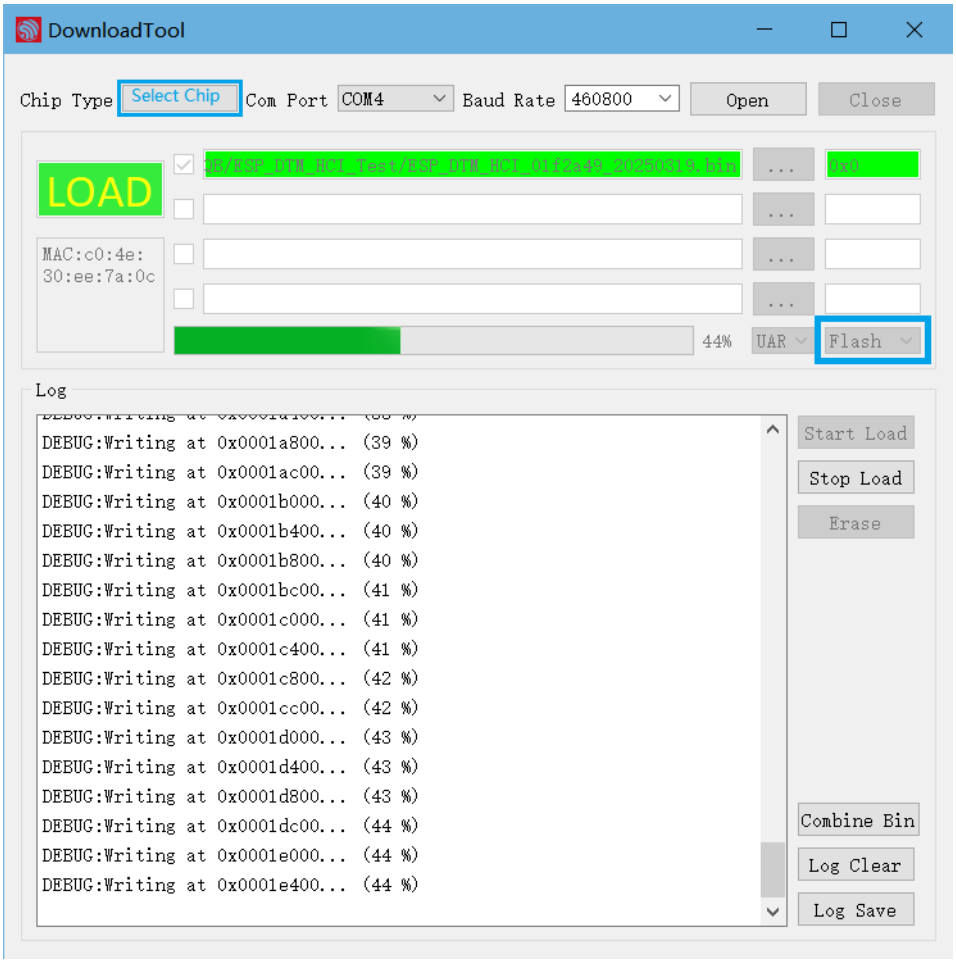


Fig. 21: Flash Firmware

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-H2 for RF debugging or testing reference.

Bluetooth LE Transmit Power Level

Table 4: ESP32-H2 Bluetooth LE Transmit Power Levels

Power Level	Bluetooth LE Transmit Power (dBm)
0	-24
1	-21
2	-18
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

3.3 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

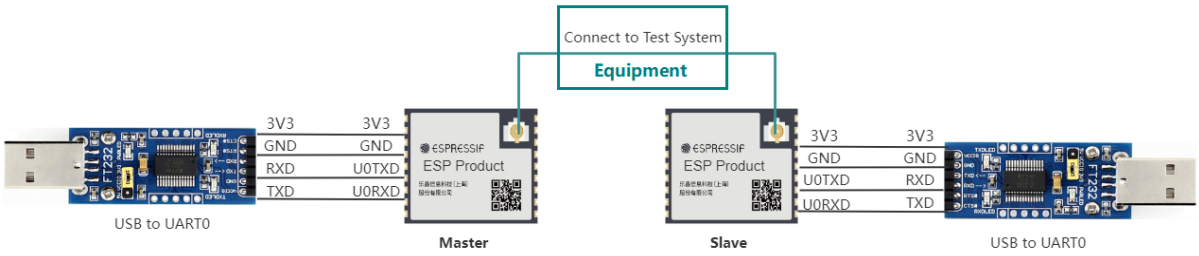


Fig. 22: Test Environment Setup

- In the test, the ESP32-H2 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-H2 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 ESP32-H2 Bluetooth LE Adaptivity Test Firmware to `0x0` via `UART`.

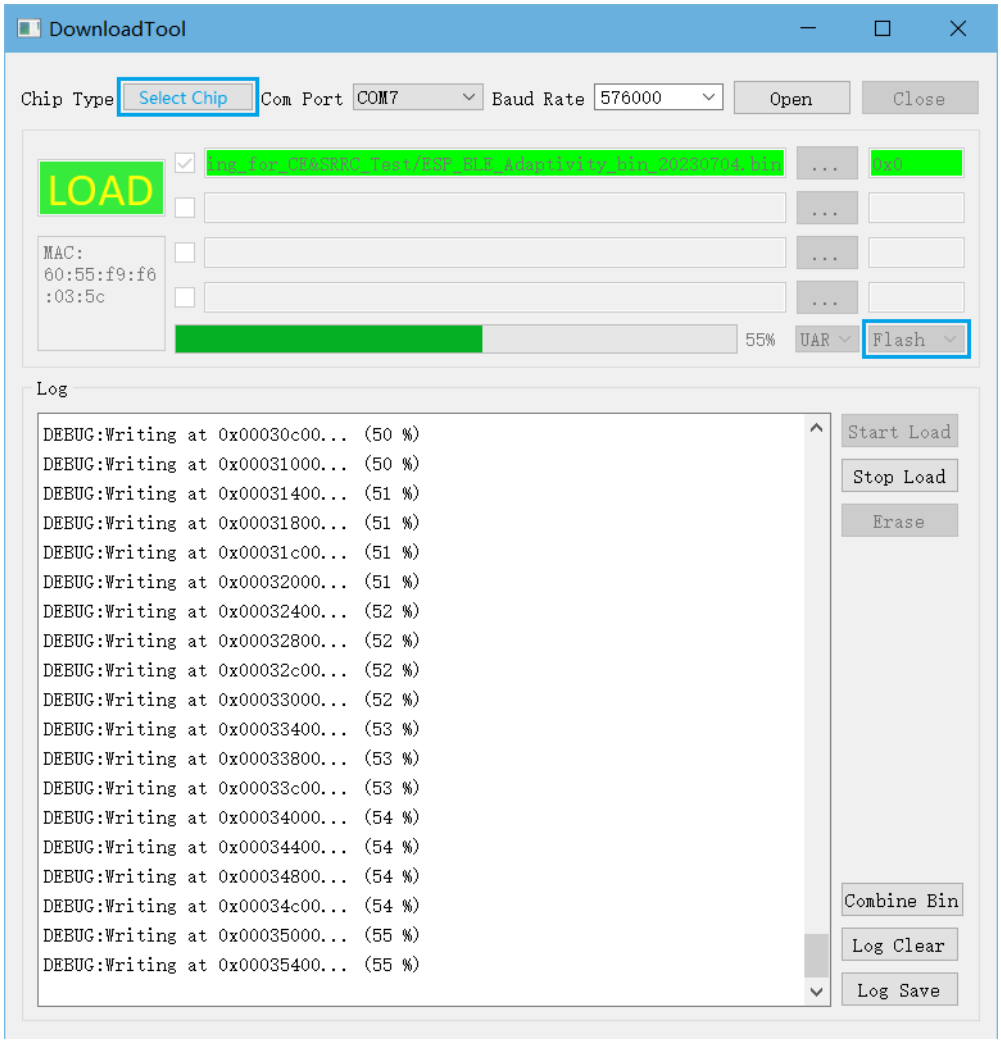


Fig. 23: Flash Firmware

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 Mater 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
bleadve -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.

3.4 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*.

3.5 802.15.4 Non-Signaling Test

802.15.4 Non-Signaling Test directly controls the device to transmit specific signals without requiring a network connection. It evaluates key performance metrics such as transmit power, spectrum characteristics, and error rate, ensuring reliable communication quality in IoT applications.

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.

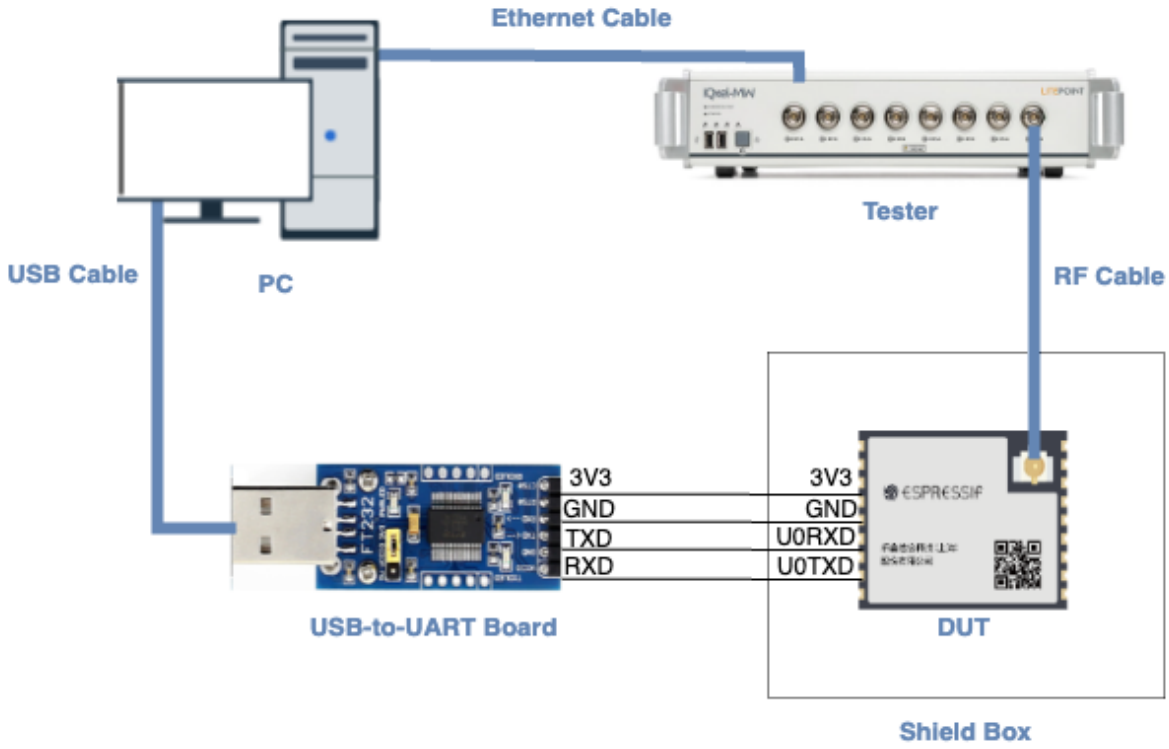


Fig. 24: Test Environment Setup

- **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 EspRFTestTool 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-H2 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-H2 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.

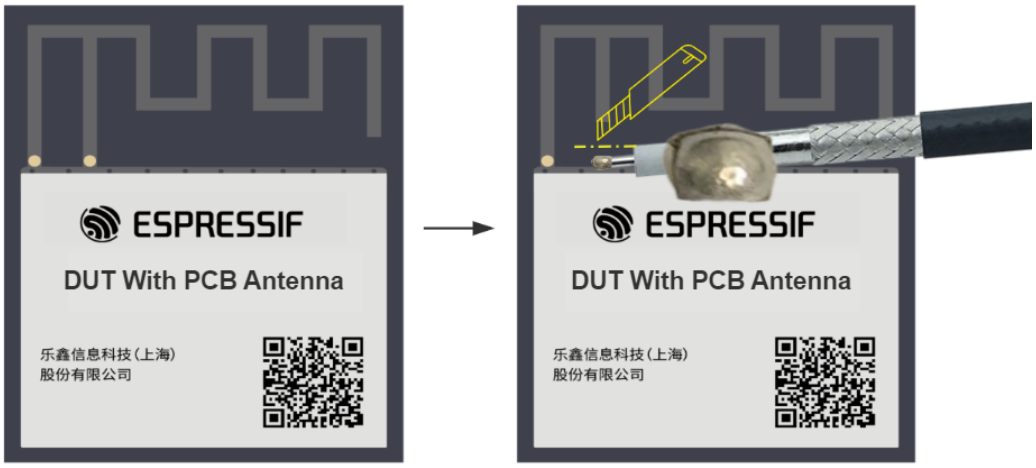


Fig. 25: Soldering RF Connection Cable to Module with Onboard PCB Antenna

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 ESP32-H2 RF Non-Signaling Test Firmware to `Flash` via `UART`.

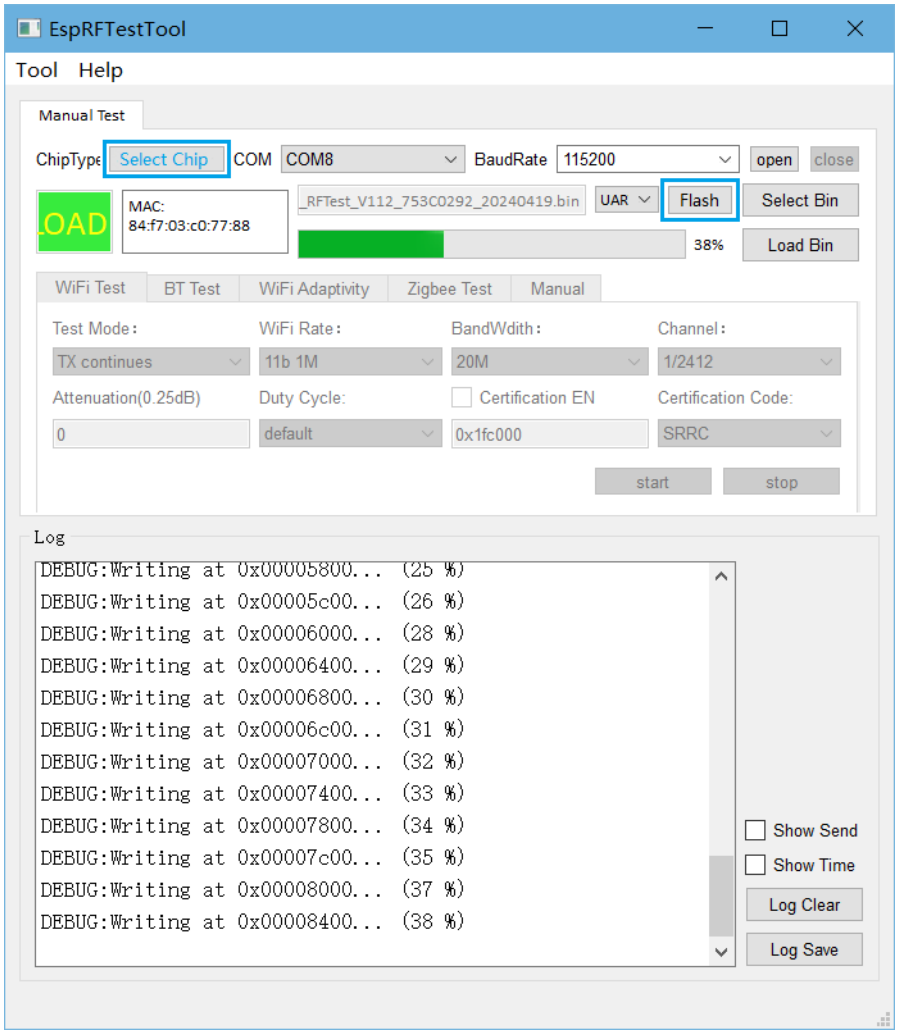


Fig. 26: ESPRFTestTool Configuration

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-H2 to 0x0.

Start Testing

Zigbee TX Performance Test

- **Test Mode**:
  - ZB TX packet: Used for TX performance tests;
  - ZB TX continue: High packet duty cycle used for certification testing.
- **Power Level**: Set the Zigbee TX power level, supporting 0~15 levels for testing.
- **Channel**: Set the Zigbee test channel.
- **Payload Length**: Set the payload length. Manual input is supported. Range: 3~127. Default: 127.

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

```
ZB TX start: len=127, chan=11, pwr=15, tx_num=0, contin_en=0
```

This indicates that Zigbee is sending packets normally, and the TX performance can be detected using the tester.

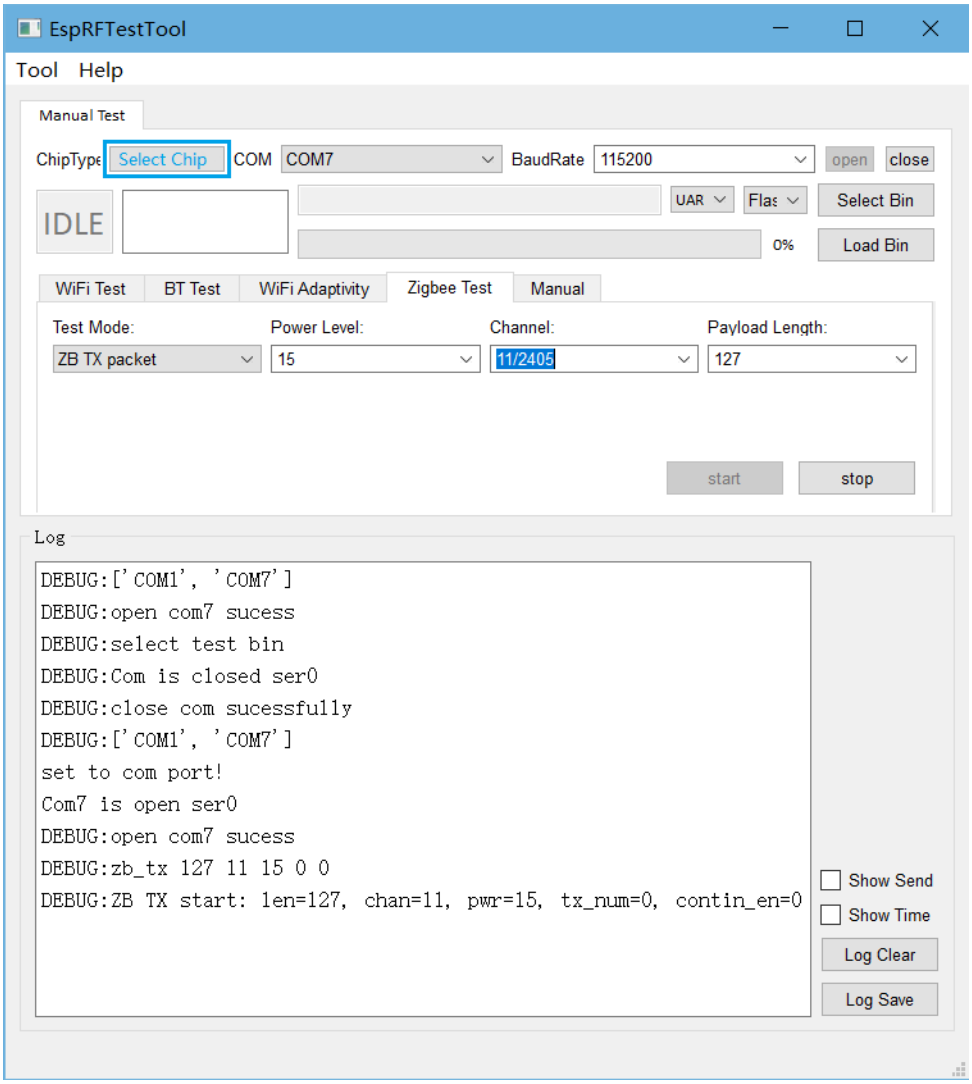


Fig. 27: Zigbee TX Performance Test

Zigbee RX Performance Test

- **Test Mode**: Set to ZB RX for Zigbee RX performance test.
- **Channel**: Set the Zigbee test channel.

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

```
RX 1000 1 1 0 0 0 -60058 0 -21398 38679
```

Among them:

- The first parameter Res[0] returns the string "RX".
- The second parameter Res[1] (decimal) indicates the number of packets received at the corresponding rate in this test. In this test, Res[1] is 1000.
- The fourth last parameter, Res[7] (in decimal), represents the total RSSI of the packets received at the corresponding rate in this test. In this test, Res[7] is -60058.

Based on the above parameters, you can calculate:

- Packet loss rate PER = [1-(Res[1]/Sent_Packet_Numbers)]*100%<=1%

- RSSI per packet = Res[7]/(Res[1])

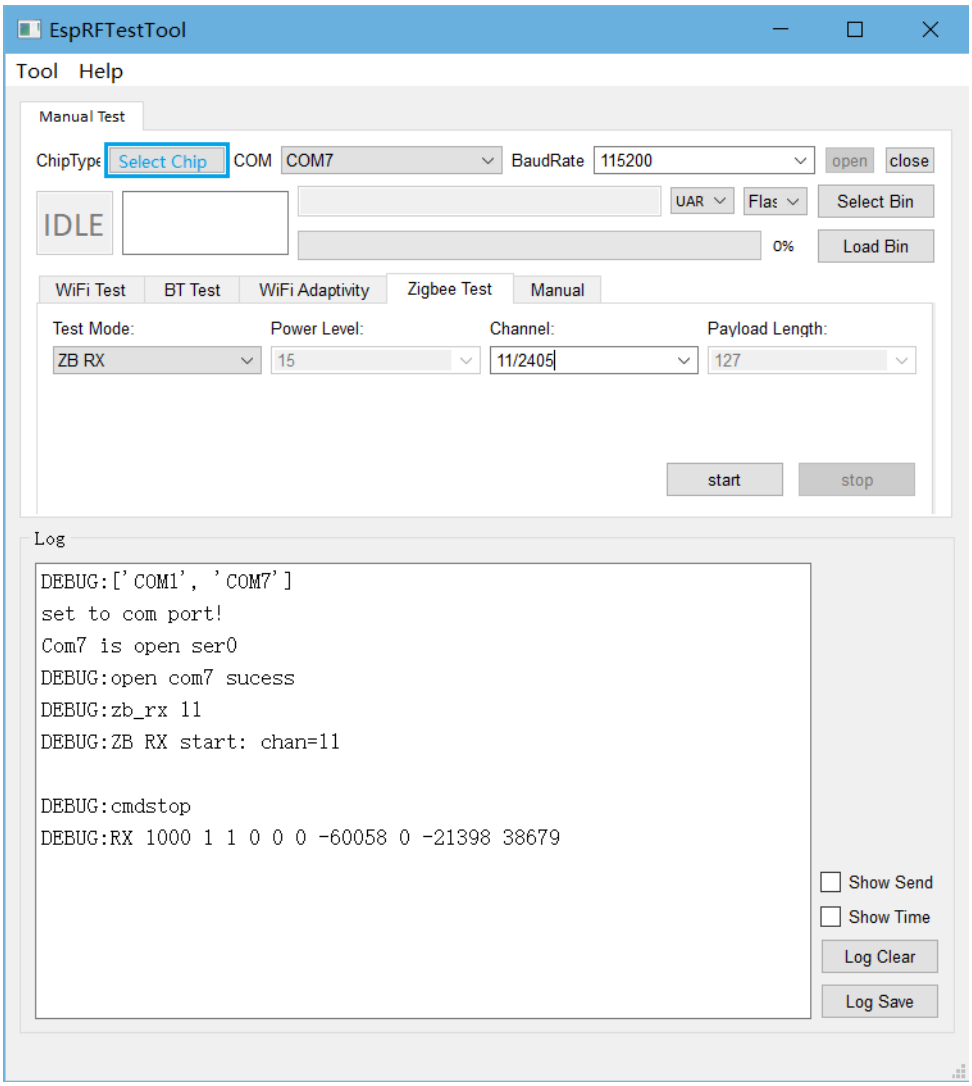


Fig. 28: Zigbee RX Performance Test

Appendix

This appendix is mainly used to explain the output target power of ESP32-H2 802.15.4, which is used for RF debugging or test reference.

Table 5: ESP32-H2 802.15.4 TX Power Level

Power Level	802.15.4 Power (dBm)
0	-24
1	-21
2	-18
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

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:

- *Bluetooth LE Non-Signaling Test*
- *Bluetooth LE DTM Test*
- *Bluetooth LE Blocking Test*
- *Bluetooth LE Adaptivity Test*
- *802.15.4 Non-Signaling 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:

- *Bluetooth LE Non-Signaling Test*
- *802.15.4 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:

- *Bluetooth LE Non-Signaling Test*
- *802.15.4 Non-Signaling Test*

5 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.
- *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.

6 Flash Download Tool User Guide

6.1 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`

6.2 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:

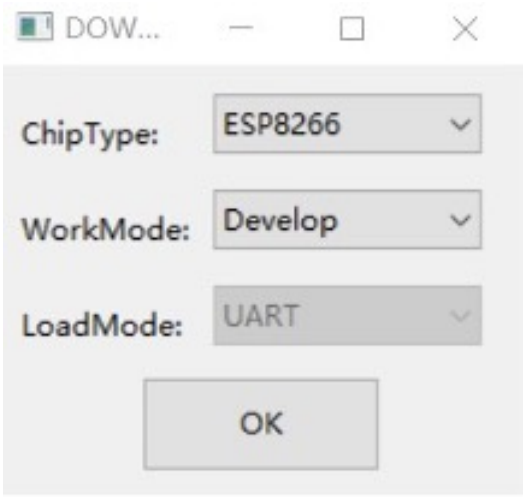


Fig. 29: Flash Download Tool Main Interface

- `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

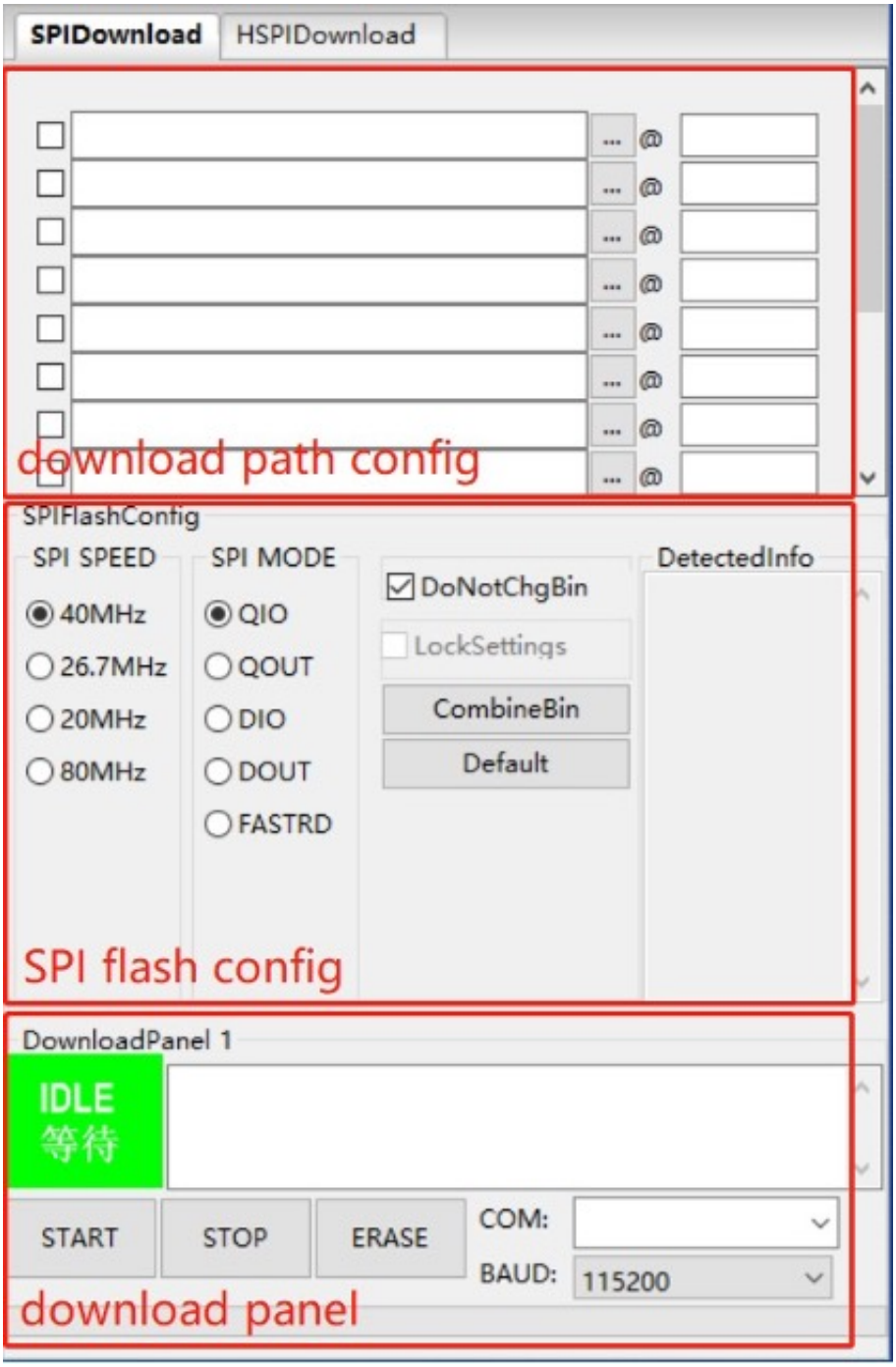


Fig. 30: SPIDownload Tab

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.)

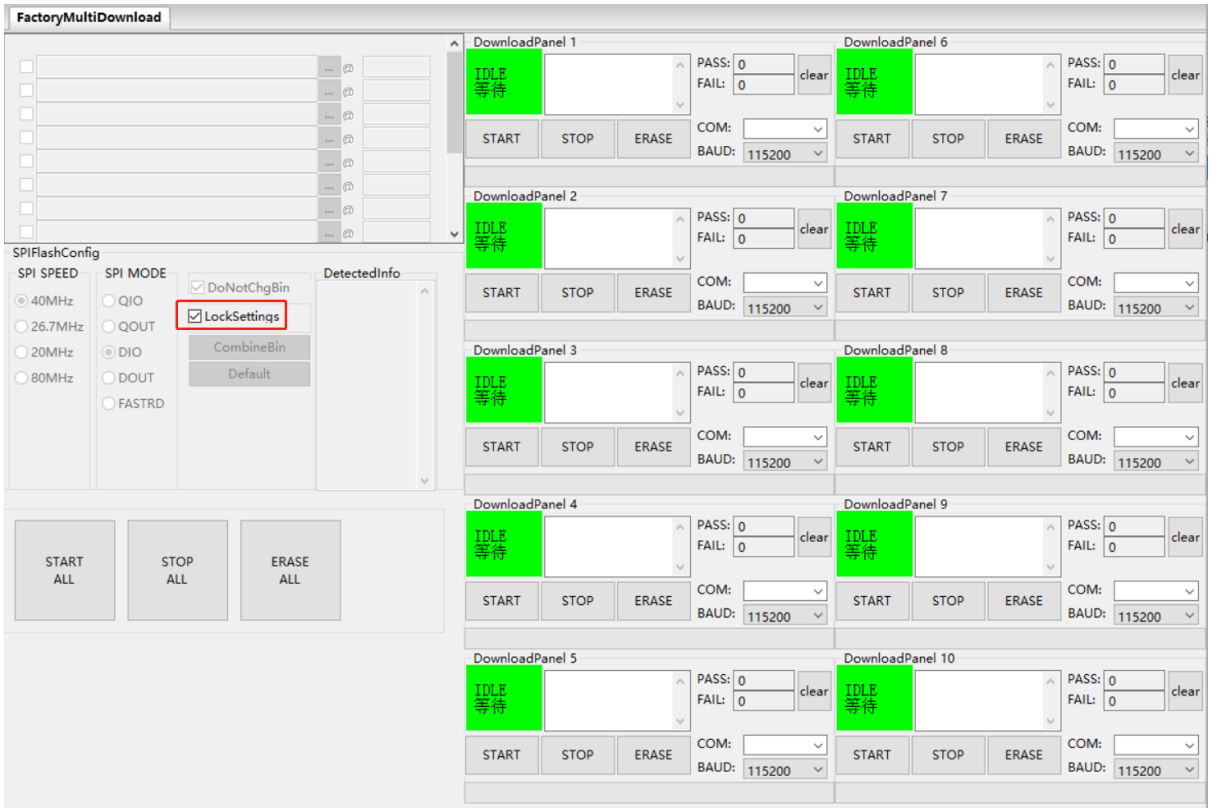


Fig. 31: FactoryMultiDownload Tab

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.

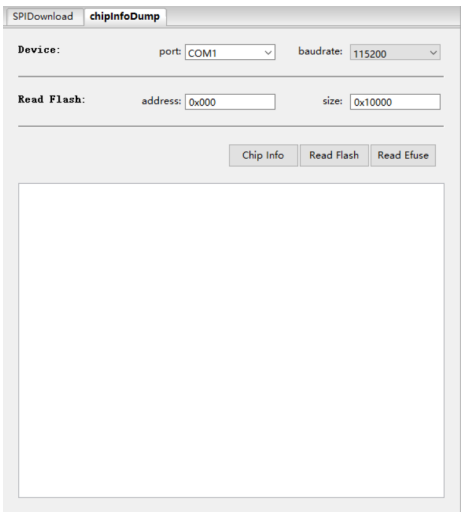


Fig. 32: chipInfoDump Tab

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

Note:

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

6.3 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-H2 only supports regular download, and will support encrypted download later.

Regular Download

1. Pull GPIO9 low and GPIO8 high to enter the downloading mode.
2. Open the download tool, set `ChipType` to `ESP32`, `WorkMode` to `Develop`, and `LoadMode` to `UART` as shown in the figure below. Then, click `OK`

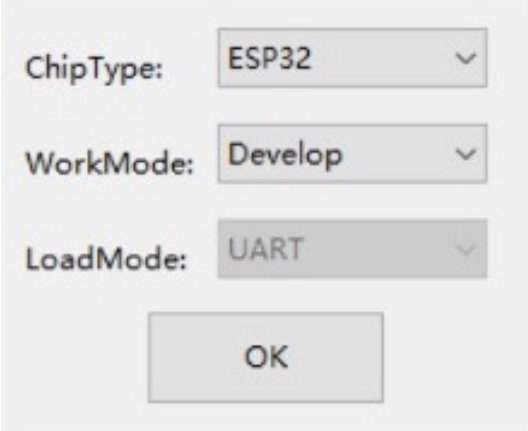


Fig. 33: Selecting Device —ESP32 Download Tool

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.

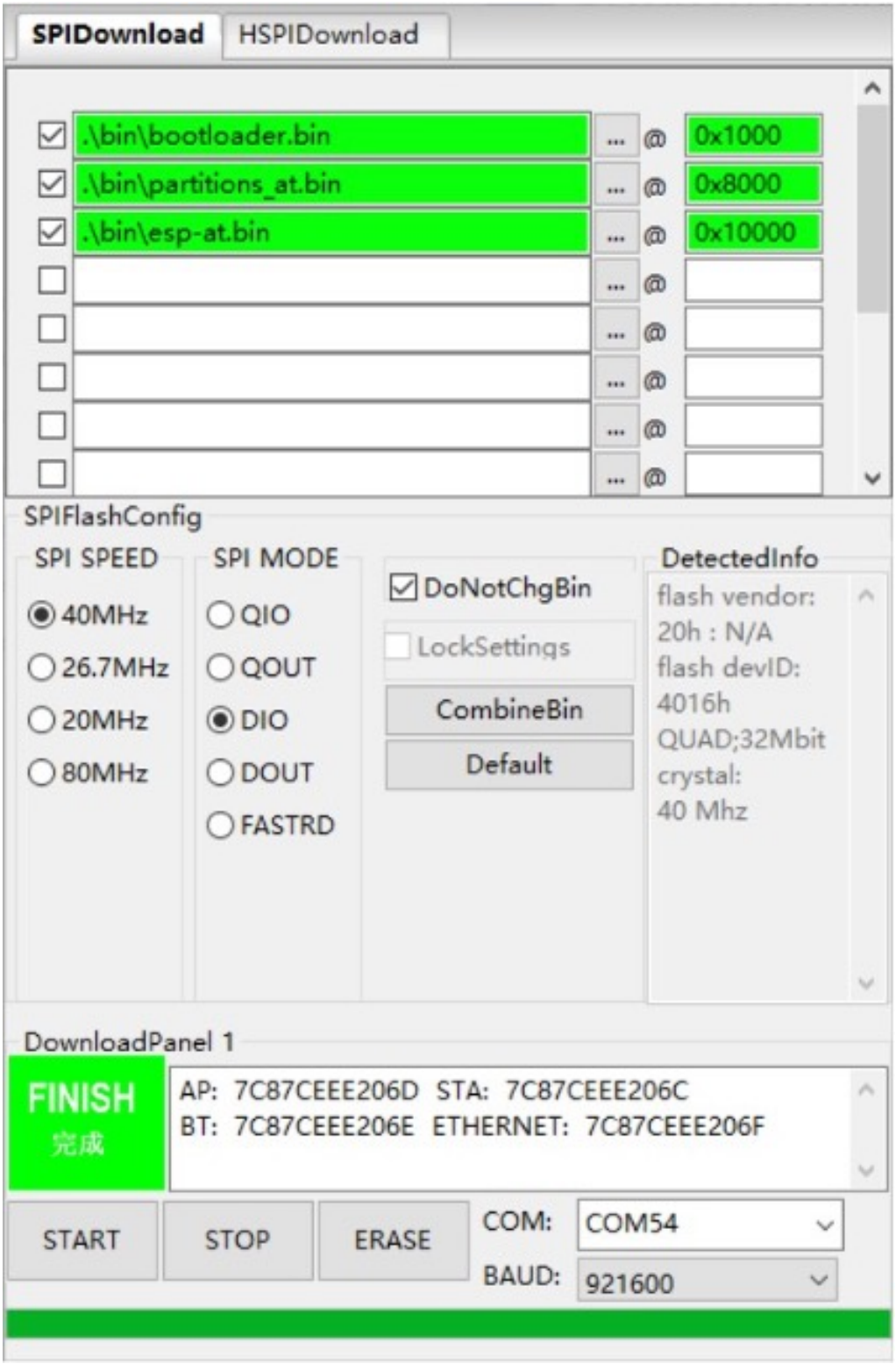


Fig. 34: Download Completed

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.

Table 6: [ESP32-H* DISABLE FUNC] Config Option

[ESP32-H* DISABLE FUNC] Config Option	Description
dis_direct_boot = False	Configures whether to disable direct boot
soft_dis_jtag = False	Configures whether to soft-disable JTAG
dis_pad_jtag = False	Configures whether to hard-disable JTAG
dis_usb_jtag = False	Configures whether to disable USB JTAG

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

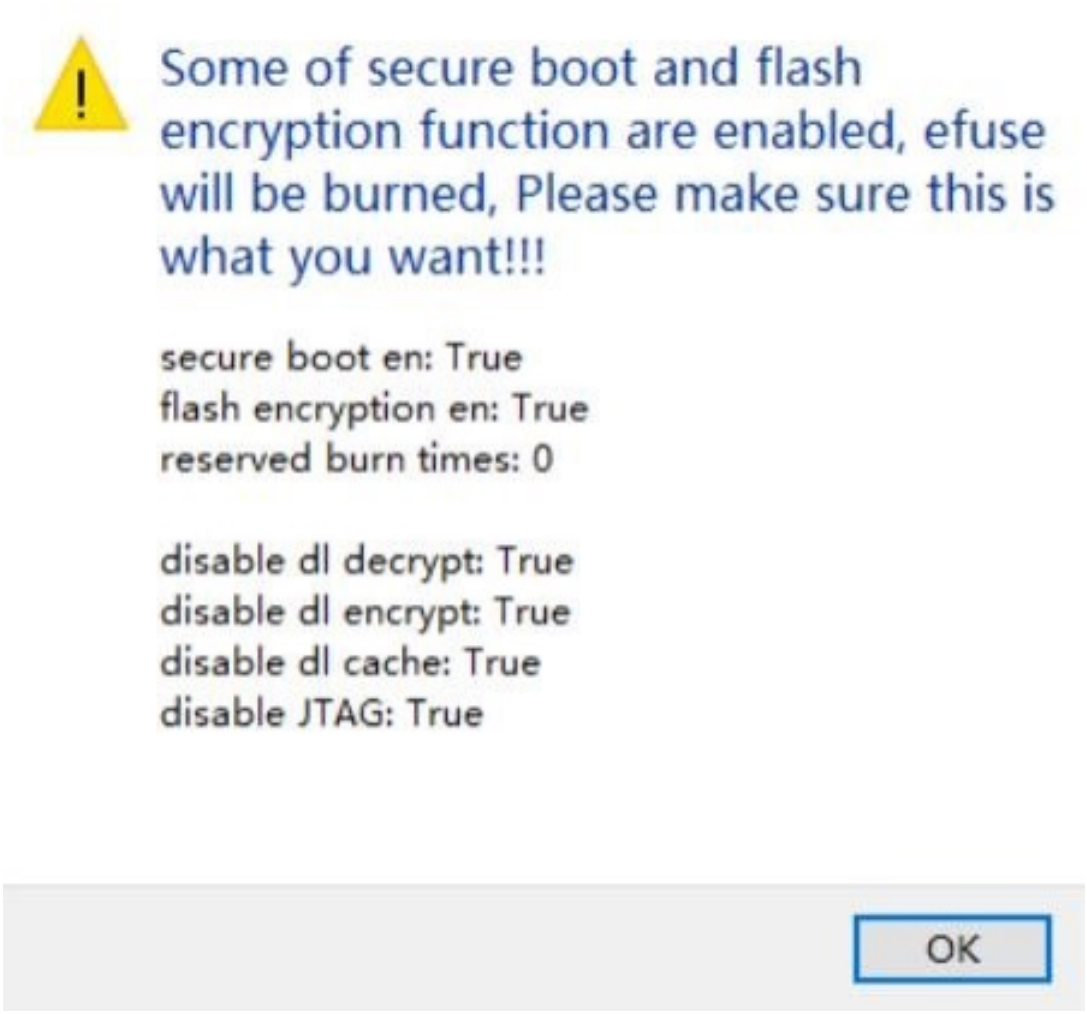


Fig. 35: ESP32 Prompt Message of Enabling Flash Encryption and Secure Boot

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.

7 Test Fixture Manufacturing Instruction

7.1 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.

7.2 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:

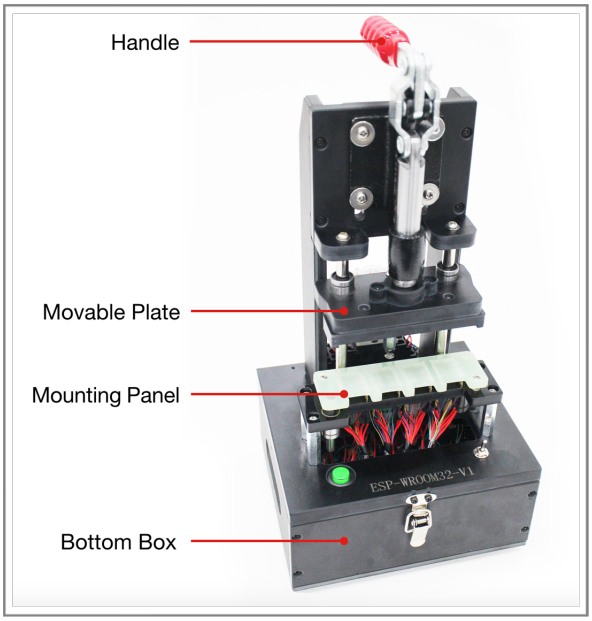


Fig. 36: The Structure of a Typical Test Fixture (ESP-WROVER)

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 7: 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.

7.3 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:

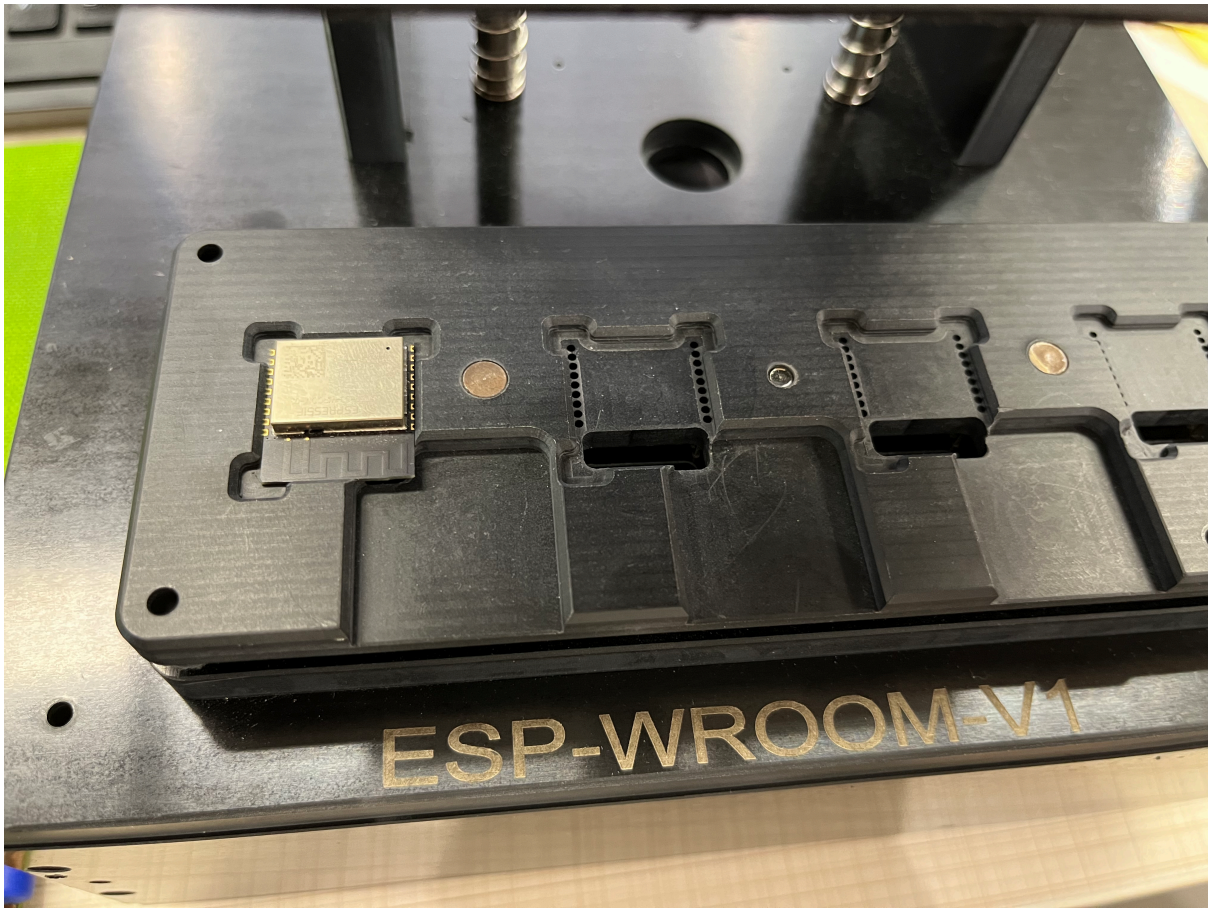


Fig. 37: The Mounting Panel for the ESP-WROVER Test Fixture

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. 38: 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. 39: 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:

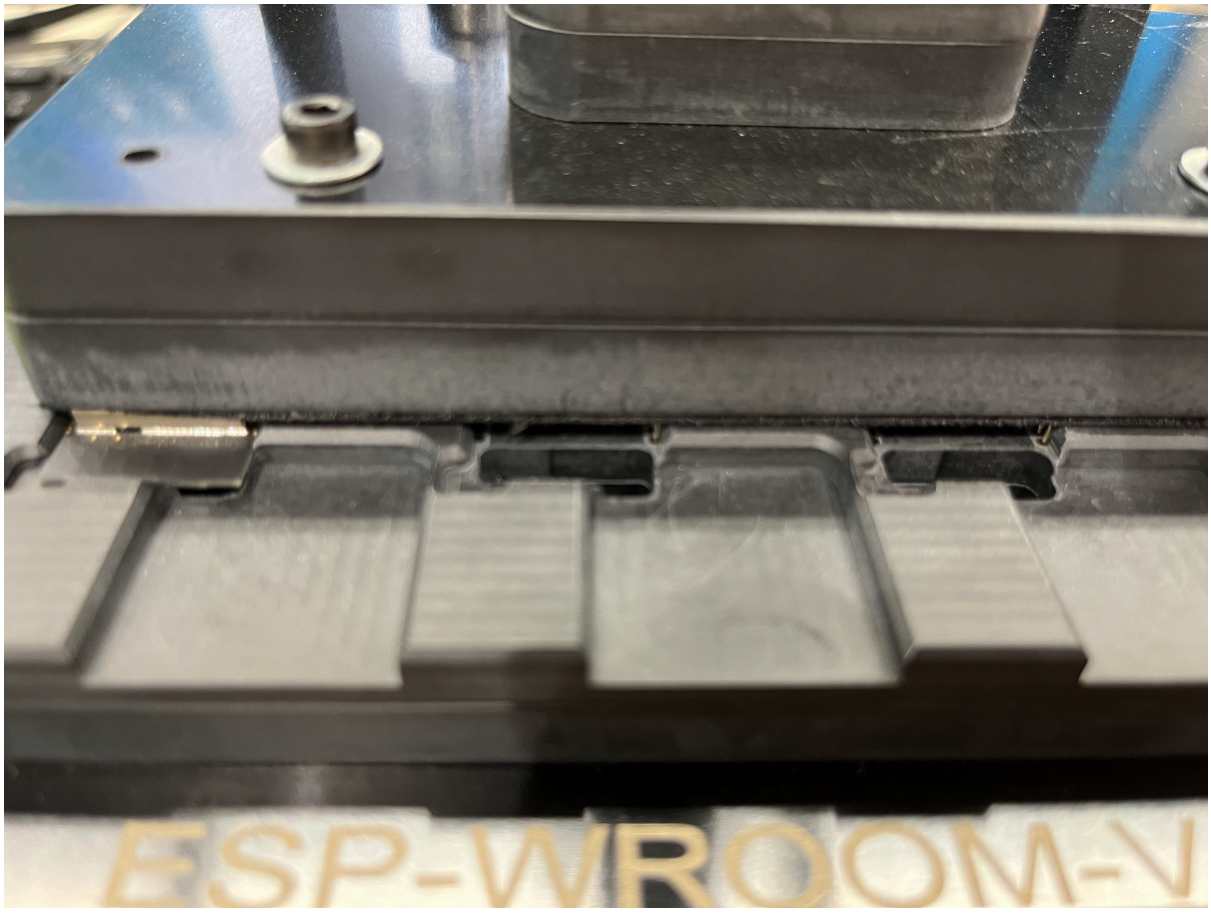


Fig. 40: The Movable Plate and the Mounting Panel

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:

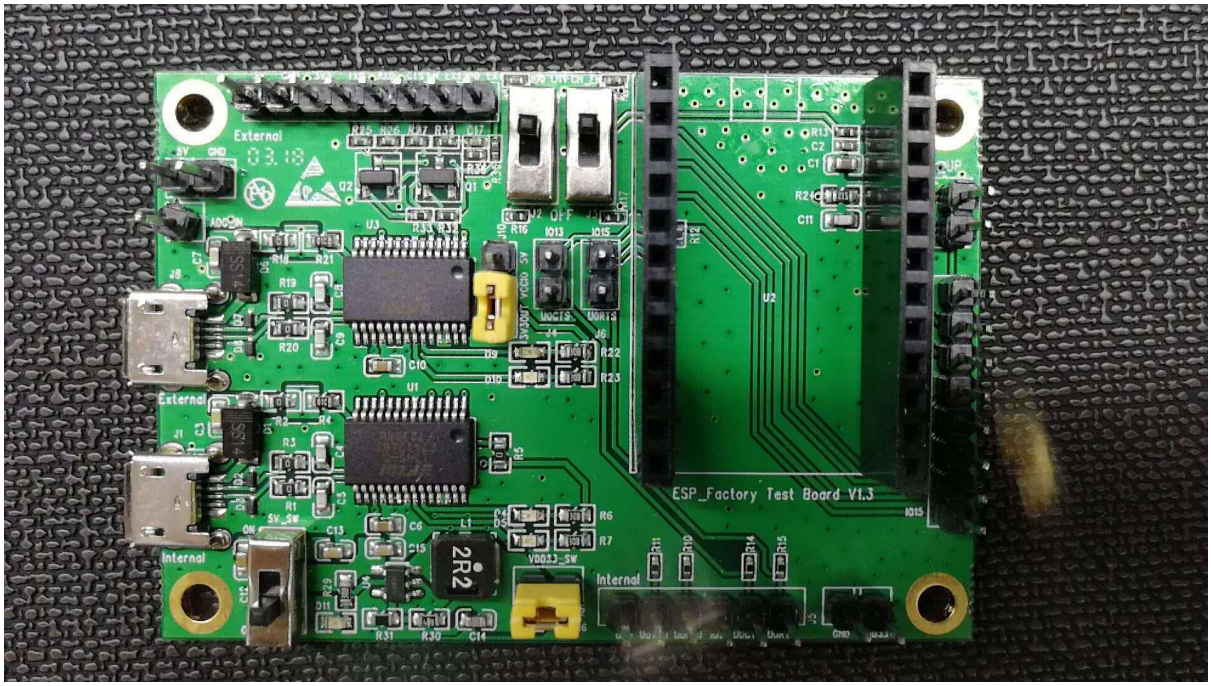


Fig. 41: A Typical Serial Port Board (ESP_Factory Test Board V1.3)

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 8: 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 9: 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).	1. Press the handle, so that the switch located in positions 1 and 2 is turned on and the serial port boards are powered on; 2. The DIP switch located in position 3 and 4 is used to control the conductivity of the wiring, and therefore the working modes.
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.	–

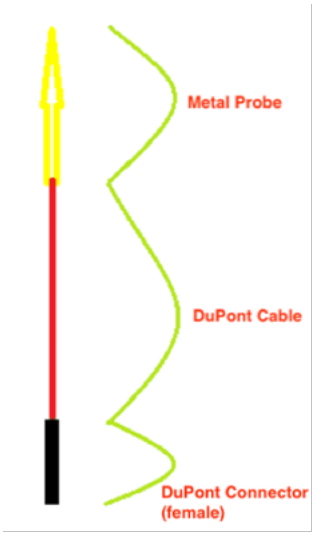


Fig. 42: A Diagram Showing How to Connect the Red Wires

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

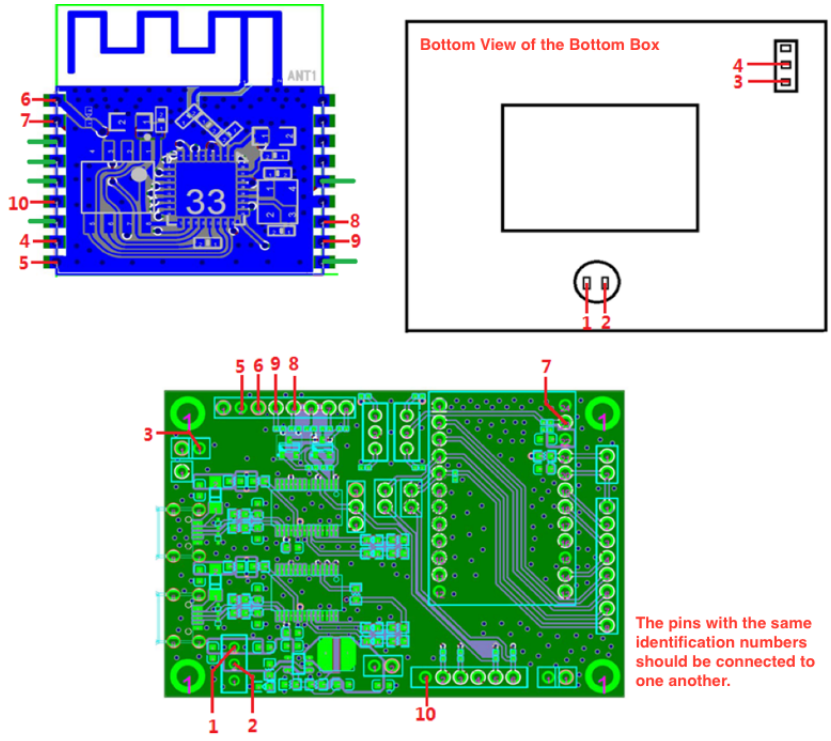


Fig. 43: The Wiring of the ESP-WROOM-02 Fixture

2. ESP-WROOM-32

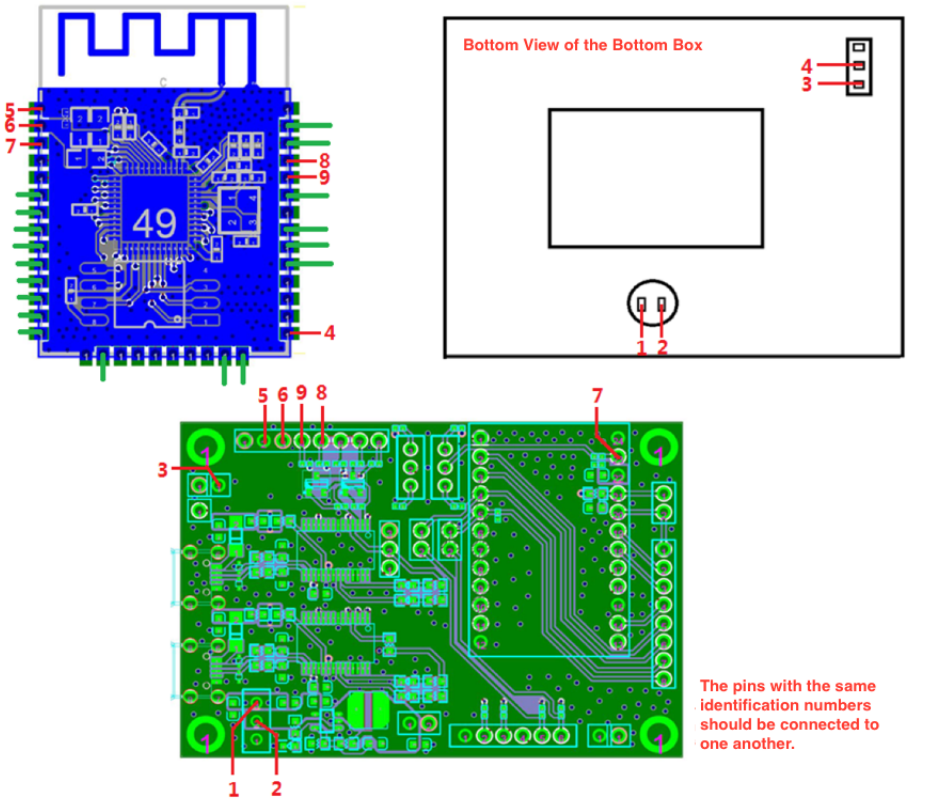


Fig. 44: The Wiring of the ESP32-WROOM-32 Fixture

3. ESP-WROVER

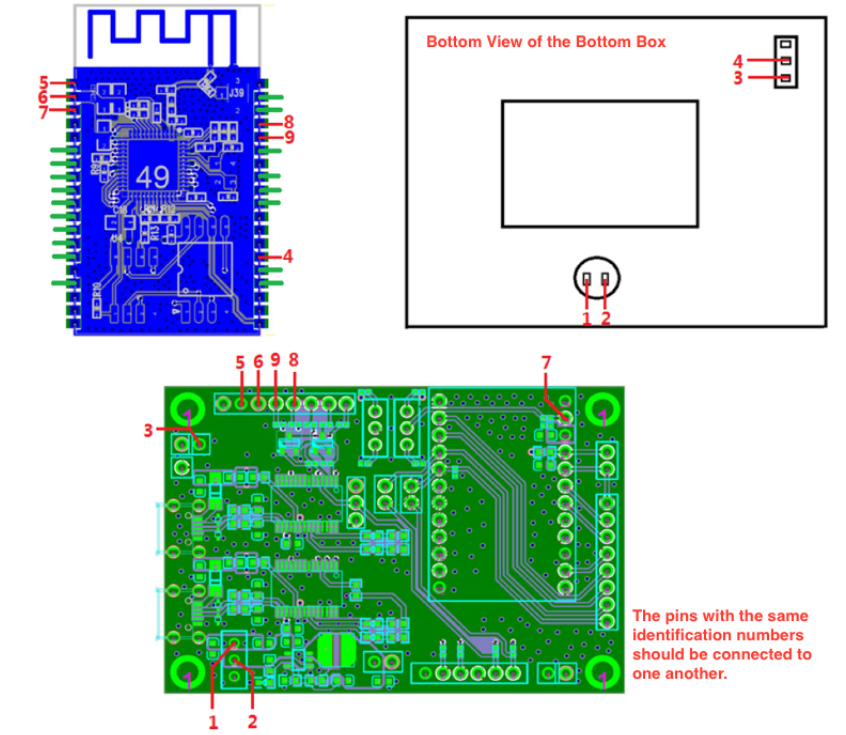


Fig. 45: The Wiring of the ESP-WROVER Fixture

When the Automatic Mode Switching on the Tool Side is supported

1. ESP-WROOM-02

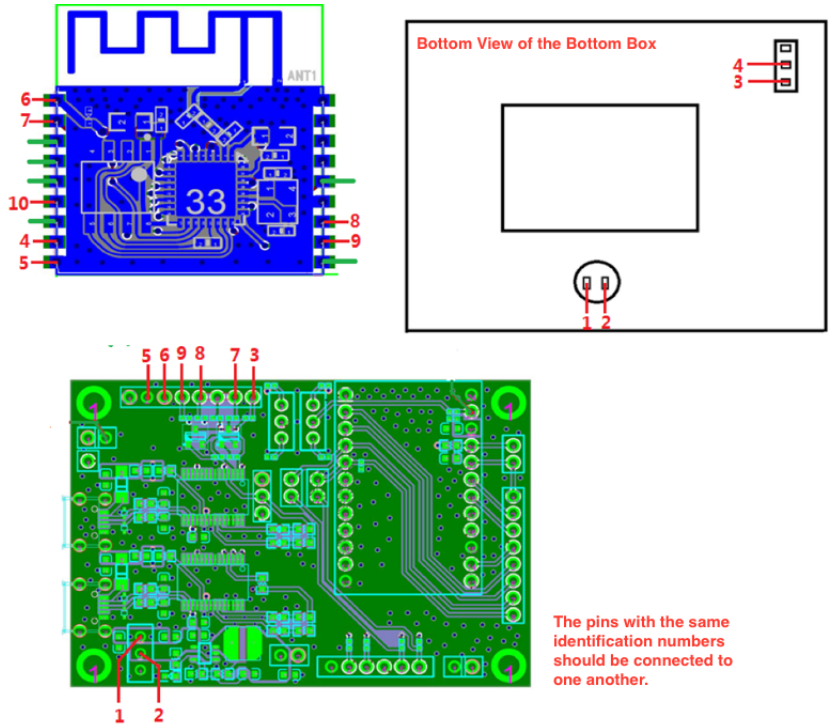


Fig. 46: The Wiring of the ESP-WROOM-02 Fixture

2. ESP32-WROOM-32

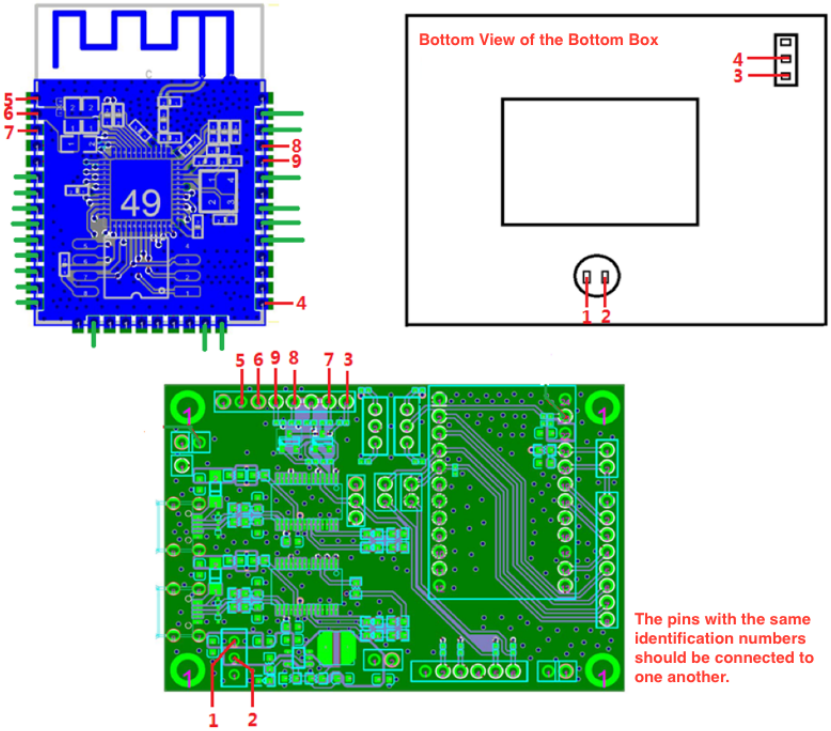


Fig. 47: The Wiring of the ESP-WROOM-32 Fixture

3. ESP-WROVER

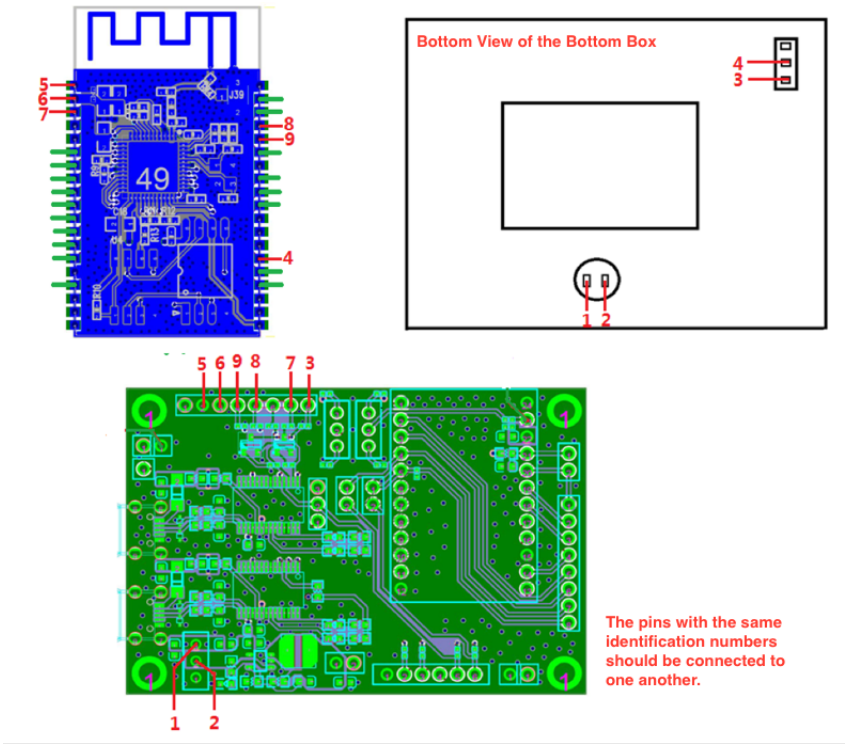


Fig. 48: The Wiring of the ESP-WROVER Fixture

7.4 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.

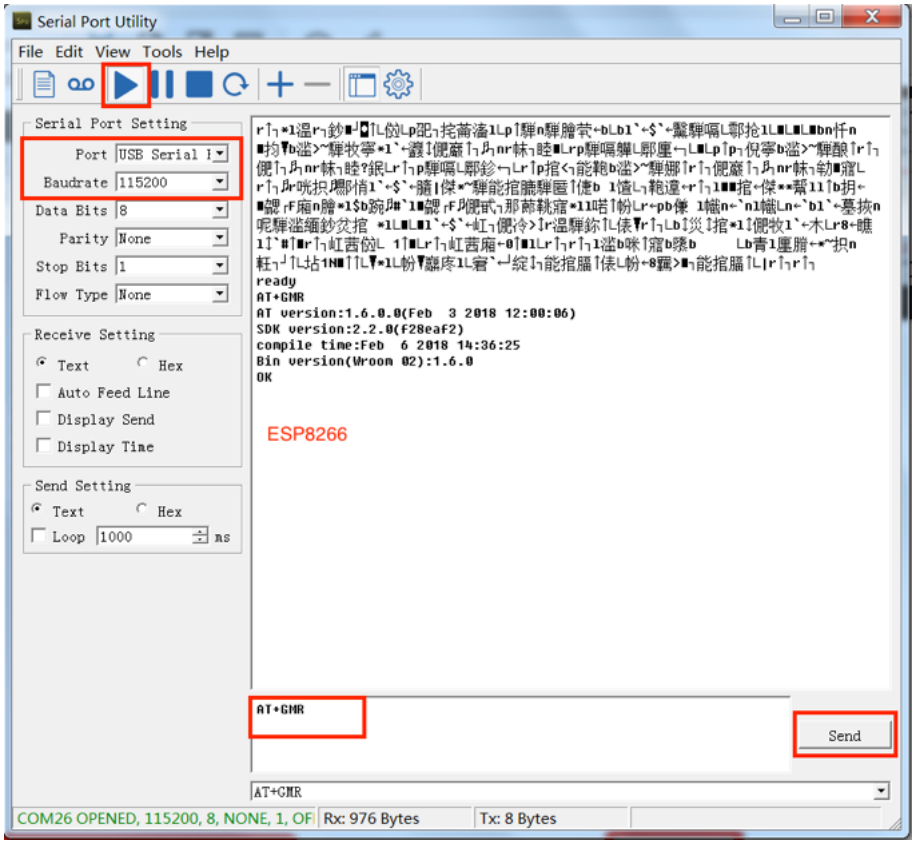


Fig. 49: AT Command Test

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.

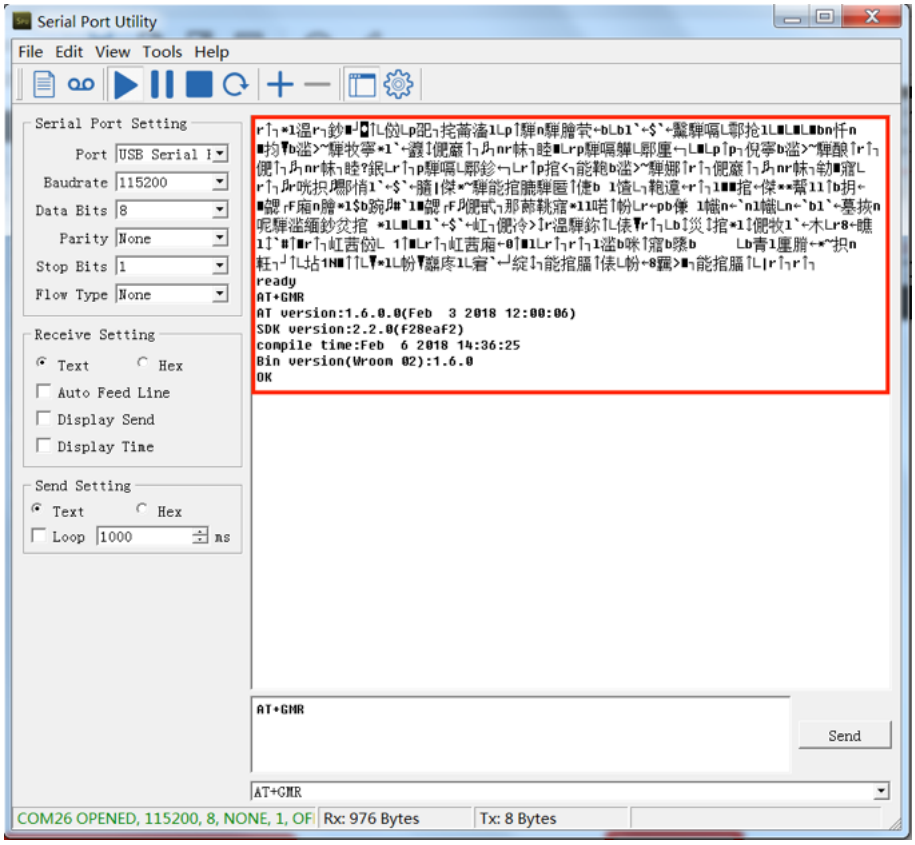


Fig. 50: Expected Result of AT Test - ESP8266 Series

- 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.

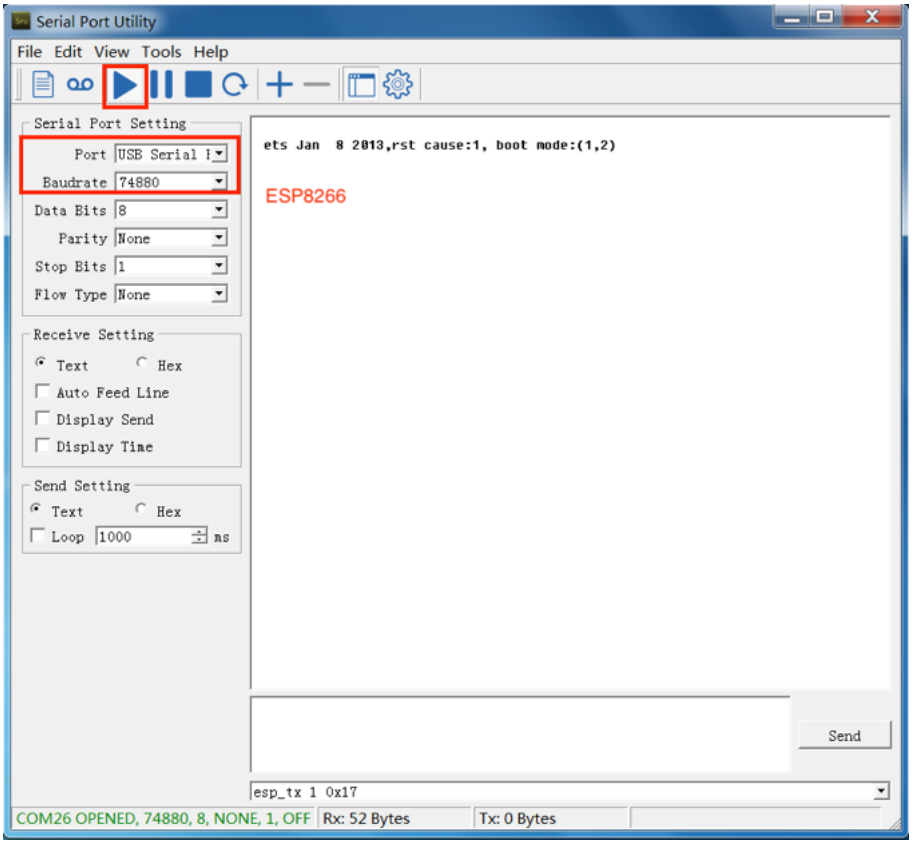


Fig. 51: Expected Result - ESP8266 Series

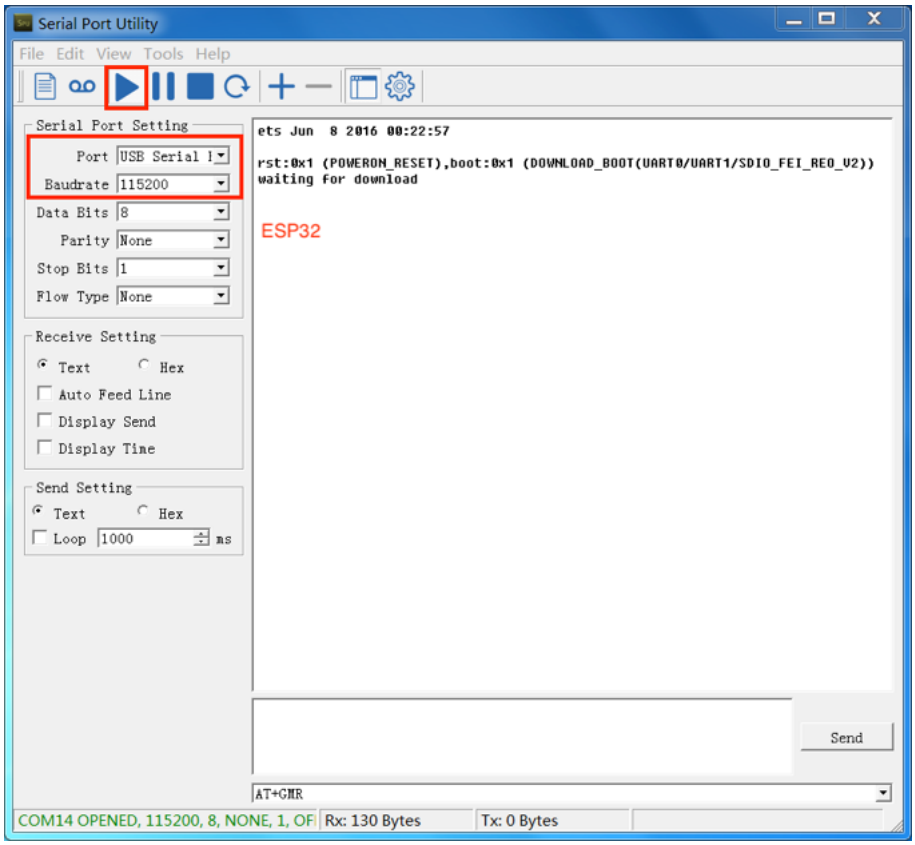


Fig. 52: Expected Result - ESP32 Series

- 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.

7.5 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 10: 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 11: 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.

8 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`

8.1 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

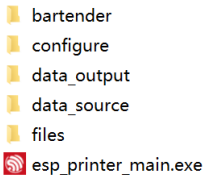


Fig. 53: QR code generator main interface (click to enlarge)

8.2 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. 54: 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.

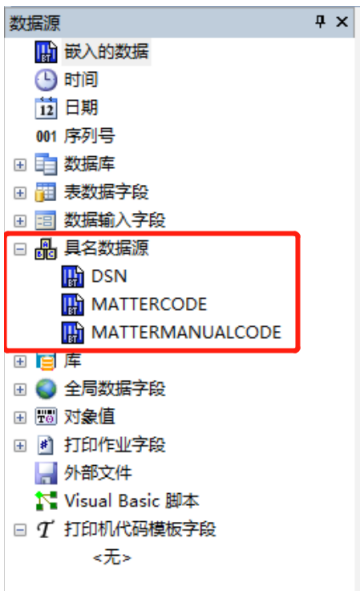


Fig. 55: Named data sources in the template (click to enlarge)

- 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	rssiLimit	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
	devBaud	115200	Scanning board baud rate
v2_scanboard (only for V2 scanning boards)	scan_timeout	Default: 10	Scan timeout
	case_command	2	Fixed value
bartender	version	[2022, 2016]	BarTender Software Version[1]

8.3 Start Printing

[1] Currently, only version 2016 and 2022 are supported.

Interface

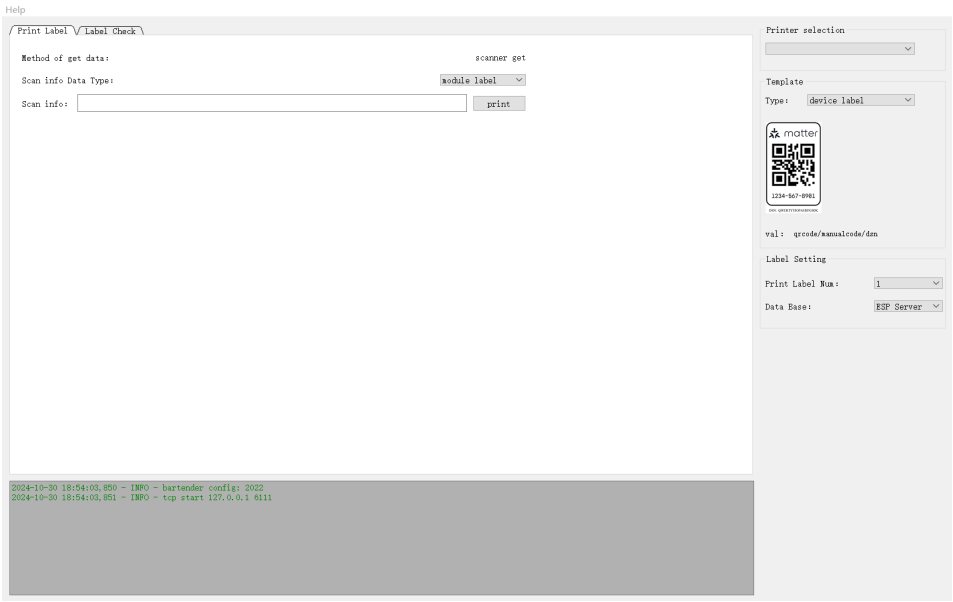


Fig. 56: Advanced Options (click to enlarge)

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. 57: 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:

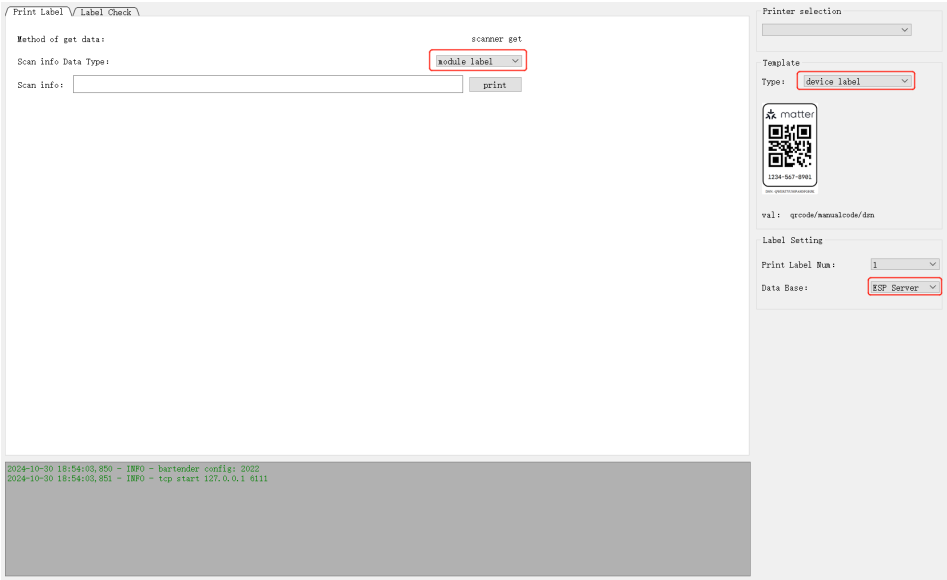


Fig. 58: Print by scanning the shield QR code (click to enlarge)

- Printing by scanning a printed label:

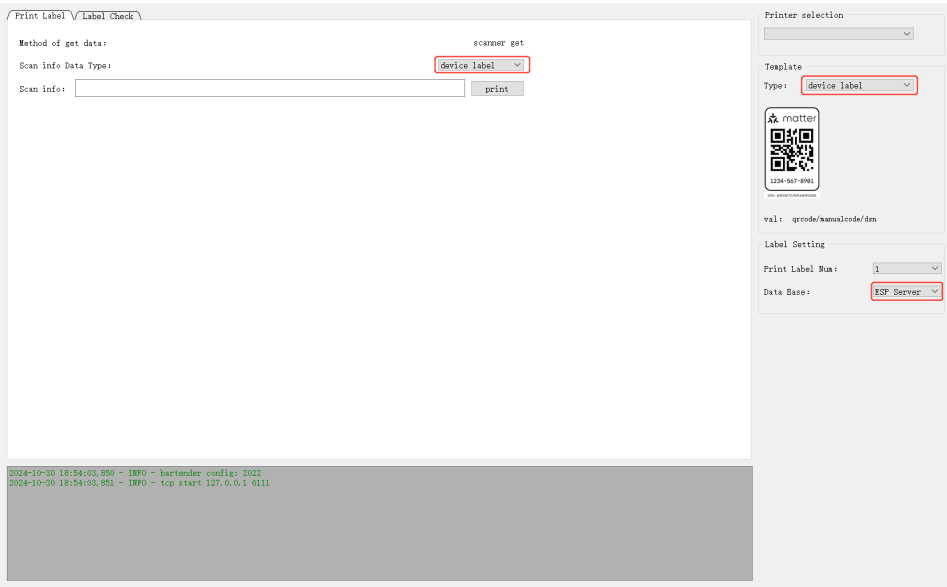


Fig. 59: Print by scanning a printed label (click to enlarge)

8.4 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.

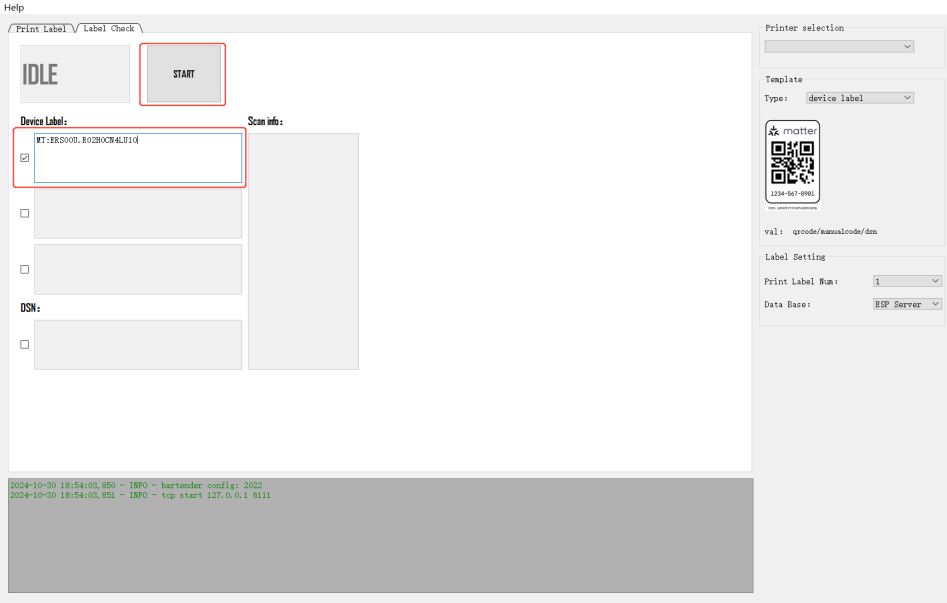


Fig. 60: QR code check (click to enlarge)

- 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.

8.5 Integrate Laser Marking

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

Configuration

Table 12: 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

8.6 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`

8.7 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. 61: Choose advanced installation options (click to enlarge)

2. Use the default installation path.

Fig. 62: Default installation path (click to enlarge)

3. The installation process is as follows.

Fig. 63: BarTender Installing (click to enlarge)

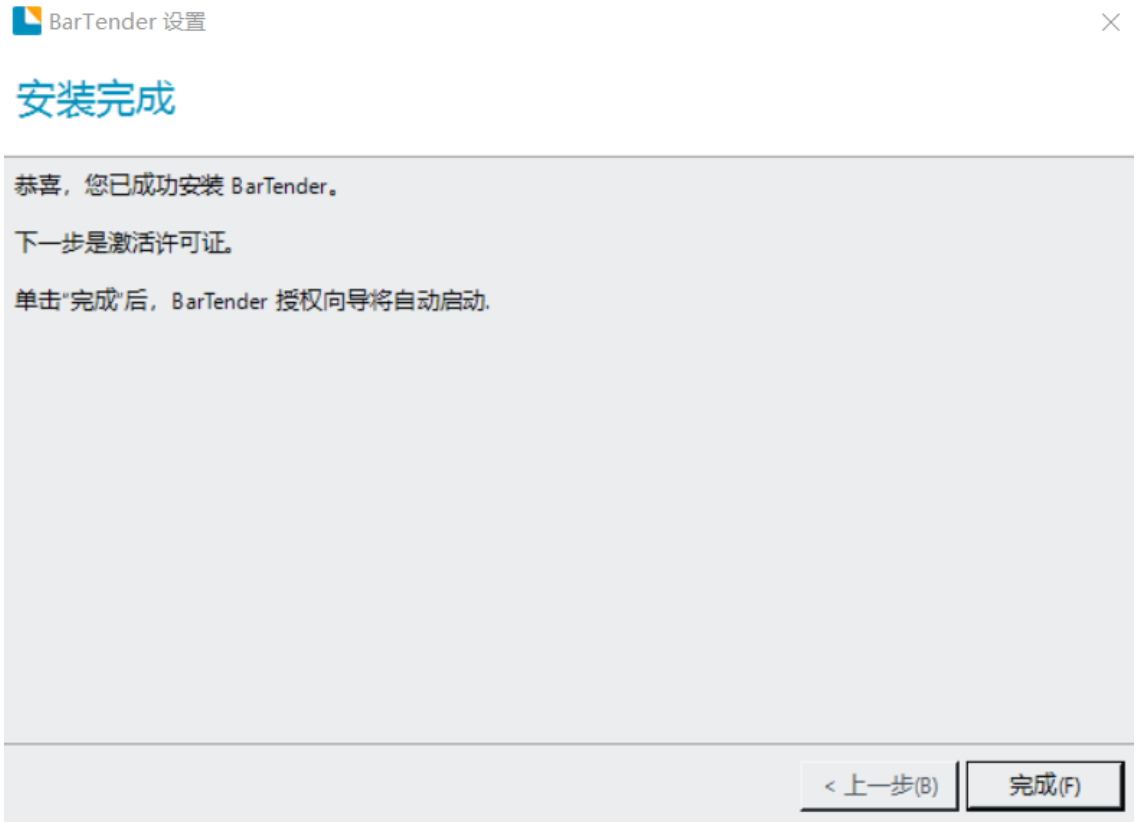


Fig. 64: Installation completed (click to enlarge)

4. Enter the serial number to activate BarTender.

Fig. 65: Enter serial number (click to enlarge)

9 FAQ

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

9.1 RF Testing

1. What should I do if the EspRFTestTool 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.

9.2 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.

9.3 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:

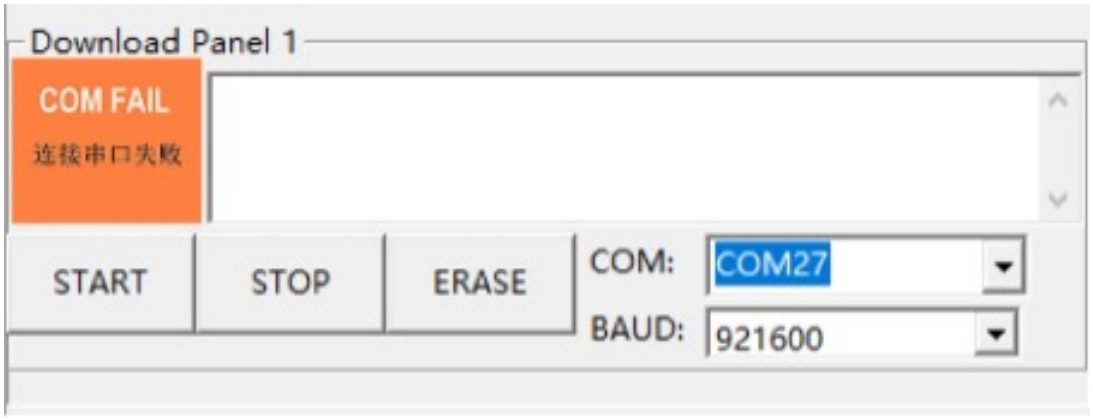


Fig. 66: Connection Failure of Serial Port

- 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?

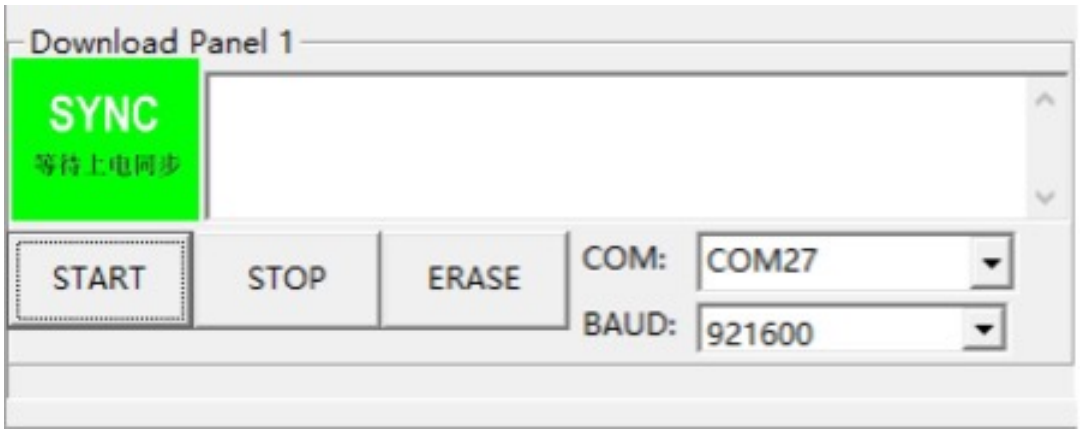


Fig. 67: Download Panel

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. 68: 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.

9.4 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.

10 Related Documentation and Resources

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

11 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.