ESP32 & ESP8266

RF Performance Test Demonstration

Version 3.0

Espressif Systems

About This Guide

Release Notes

Date	Version	Release notes
2017.12	V 1.0	Initial release.
2018.03	V 2.0	An RF test tool is used to run the tests.
2019.01	V 3.0	An adaptivity test is introduced in the RF test tool.

Documentation Change Notification

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

Certification

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

Table of Contents

1. Test Overview

1.1. Introduction

This guide demonstrates how to run an RF related performance test and certification test using an RF tester to check an ESP32 / ESP8266 based product, either functioning as a standalone controller or an integrated part of MCU.

> 📖 ***Note:***
>
> - *Download link for the test tool (open in a browser):*
>
>   *download.espressif.com/release/ESP_RFtest_and_Certification_tool.zip*
> - *Download link for the newest test bin files (open in a browser):*
>
>   *download.espressif.com/release/espressif/ESP_RFtest_and_Certification_bin.zip*
> - *For more documentation on ESP32 / ESP8266, please visit the Espressif website.*

1.2. Product Information

For the detailed information on a specific product, please check its datasheet on the *Espressif website*.

2. Test Framework

2.1. Test Setup

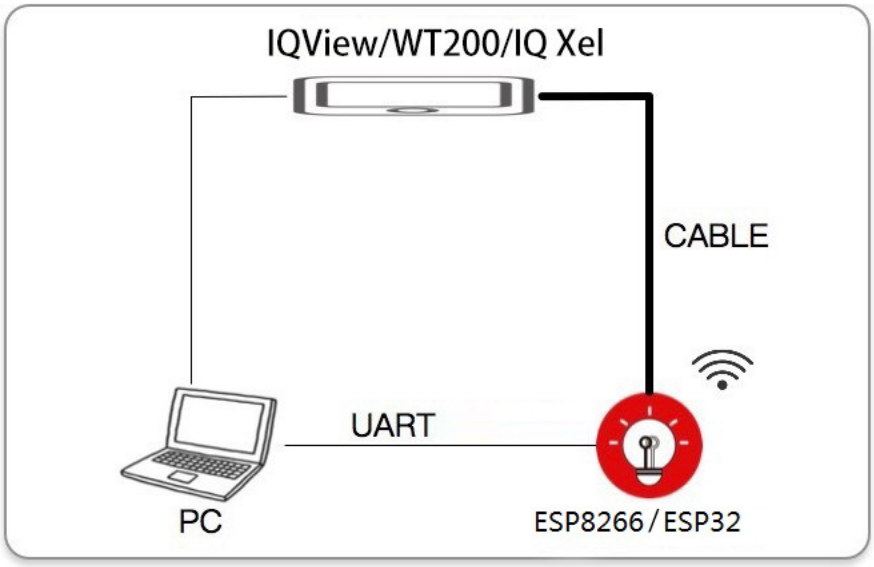


Figure 2-1. RF Test Framework

- **ESP32 / ESP8266**: An ESP32 or ESP8266 product, also referred to as a device under test (DUT) in this guide.
- **PC**: Running EspRFtestTool, an RF test tool provided by Espressif.
  - The PC communicates with a DUT through the UART interface, by which configuration-related commands specific to different test modes are sent.
- **IQView/WT200/IQ Xel**: An RF tester used for testing the RF performance of the DUT working in different modes.

2.2. Test Process

1. To test a DUT, the following pins should be led out:
   - For ESP32: 3V3, EN, GND, U0TX, U0RX, and GPIO0.
   - For ESP8266: 3V3, EN, GND, U0TX, U0RX, GPIO15 (connect to GND), and GPIO0.

> 📖 ***Note:***
>
> *GPIO0 is mainly used to switch between Download mode and Flash operation mode:*
>
> - ***Download mode**: GPIO0 is connected to GND and pulled low (used for downloading bin files).*
> - ***Flash operation mode**: GPIO0 is floating or pulled high.*

2. Connect the DUT to a serial port board, and make sure the DUT enters Download mode (connect GPIO0 to GND):
   - Connect the led-out pins to the serial port board according to Step 1. For details, please refer to Section 3.2.3.
     - GPIO0 is connected to GND and pulled low.
     - EN is connected to 3V3 and pulled high.
     - GPIO15 is pulled low (only for ESP8266).
   - Connect the serial port board to a PC using a USB cable and install the corresponding driver (see Appendix A.), to ensure the serial port can be successfully identified.
3. Open the RF test tool on the PC, download the firmware and run the test:
   - The firmware can be downloaded either to RAM or flash.
   - Run the firmware, use the RF test tool to configure the test-specific TX mode for the DUT, and start the test.

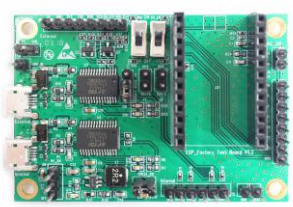
> 📖 ***Note:***
>
> *Keep in mind the difference between downloading the firmware to RAM and flash:*
>
> - ***RAM**: The firmware is ready to run immediately after it is successfully downloaded to RAM. However, the RAM program gets erased as soon as you reboot the DUT, so you will have to download the firmware again.*
> - ***Flash**: The firmware is not ready to run after the downloading process is completed. You need to switch to Flash operation mode by manually disconnecting the GPIO0 pin and re-power the DUT.*

3. Environment Setup

3.1. Related Tools

3.1.1. Hardware

Table 3-1. Hardware Preparation

Name	Picture	Quantity	Description
DUT based on ESP32 / ESP8266	N/A	May vary depending on test requirements	The products developed by customers based on ESP32 / ESP8266.
Serial Port Board		1	Serves as a USB converter that allows the test tool on the PC to communicate with the DUT.
Micro-USB cable		1	Connects the serial port board to the PC.
PC	N/A	1	• Runs relevant software. • Windows XP or Windows 7 is recommended.
Tester (e.g. IQView201x, a Wi-Fi general-purpose tester)	N/A	1	Used for testing the Wi-Fi performance-related parameters. You may use other similar testers such as iTest WT200.

3.1.2. Software

Table 3-2. Software Preparation

Name	Description
ft232r-usb-uart.zip	USB-UART converter driver (adaptable to the serial port board)
RF test tool	This program integrates downloading and running of the test firmware, as well as sending of the configuration-related commands, etc.

3.1.3. Serial Port Board

A serial port board is mainly used as a USB converter. You may use other similar boards. However, considering that some of them might have unstable performance, it is recommended to purchase the one shown below (if you would like to purchase the board from Espressif, please *contact us*).

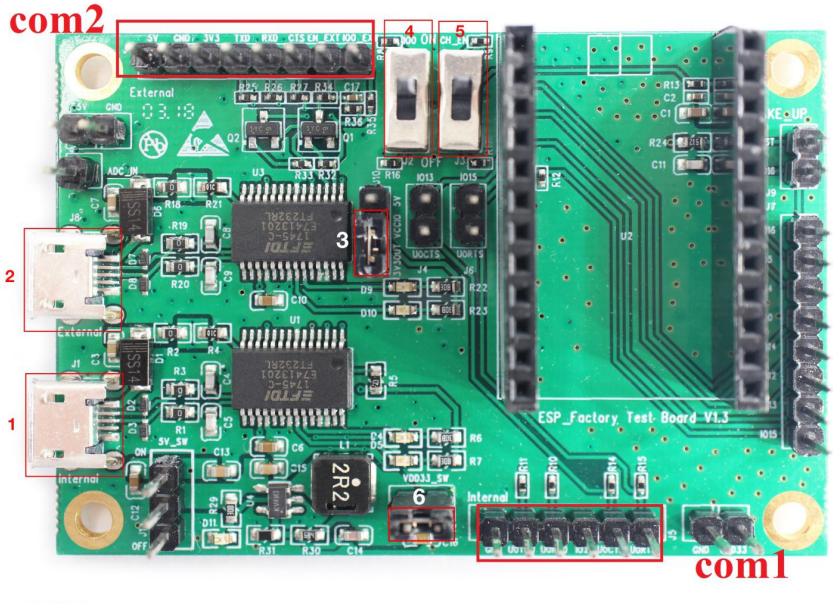


Figure 3-1. Serial Port Board

Please check your purchased board against the following criteria to make sure the switches and shorting jumpers are in their correct positions:

- Markings **com1** and **com2**: Serial ports used for communication with a PC. Markings **1** and **2** are two independent serial ports, corresponding to TX/RX/RTS/CTS.
- Marking **3**: Mainly used to jump between com2 3.3 V level and 5 V level.
- Marking **6**: Used to enable 3.3 V. Shorting jumpers should be used here.
- Marking **4**: Not used, so no need to configure.
- Marking **5**: Not used, so no need to configure.

3.2. Antenna Impedance Matching Requirement and RF Hardware Setup

3.2.1. Antenna Impedance Matching Requirement

For the electromagnetic compatibility (EMC) test, the π-type impedance matching circuit for the antenna should meet the requirements shown in Figure 3-2.

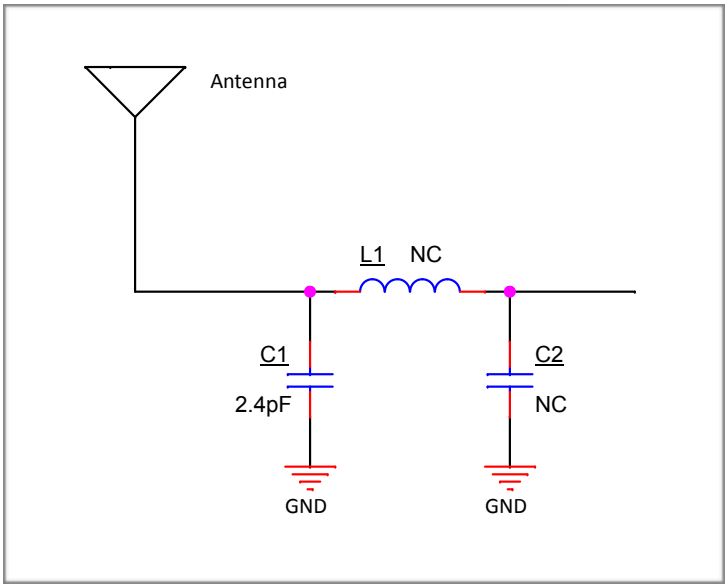


Figure 3-2. π-type Impedance Matching Circuit for the Antenna

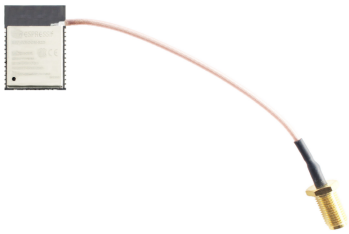
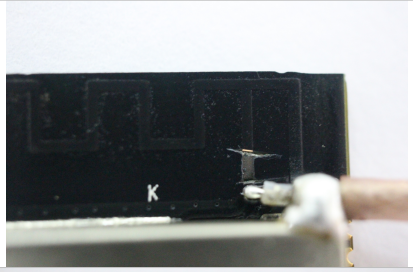
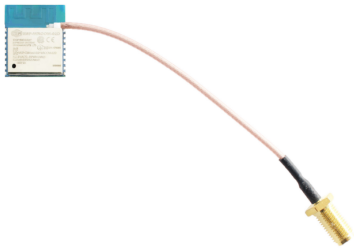
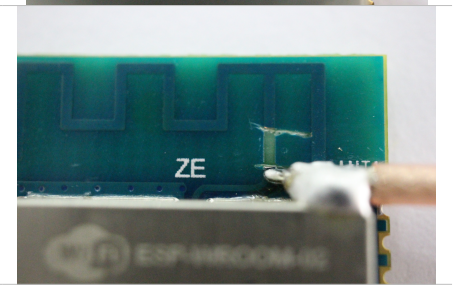
> 📖 ***Note:***
>
> *The recommended value for the capacitor C1 is 2.4 pF. L1, C2, along with C1, perform impedance matching for the antenna. The impedance value varies for different modules.*

3.2.2. RF Hardware Setup

3.2.2.1. Conduction Test

- Solder the DUT's antenna-related part to one end of a coaxial cable, and connect the cable's other end to an SMA (SubMiniature version A) connector which can later be plugged into an RF tester.
- Before soldering the cable, do not forget to cut the trace to PCB antenna, to ensure the test data is accurate.
- The test setup for ESP32 and ESP8266 modules is shown in Table 3-3 for your reference.

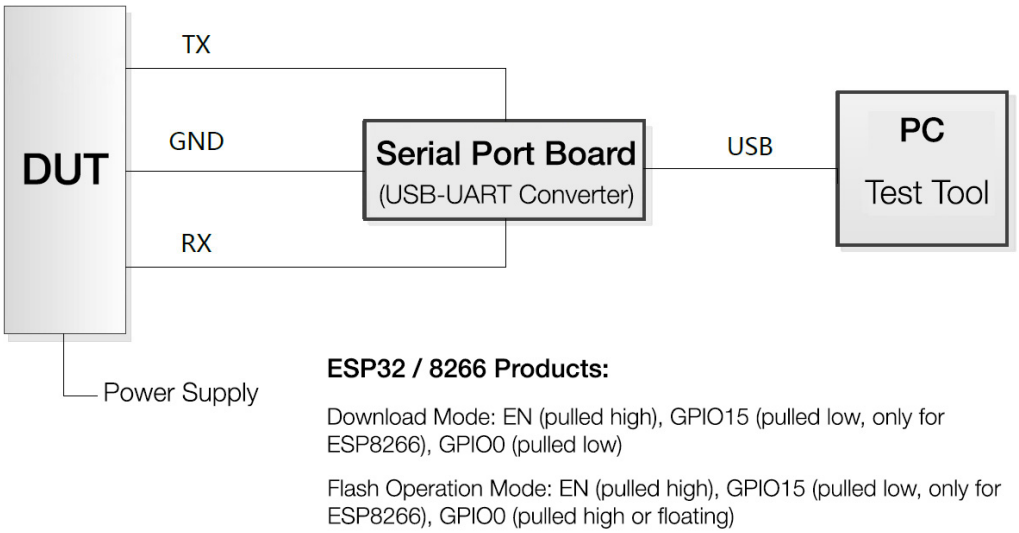
Table 3-3. Setup Demonstration for Conduction Test

	RF-related Connection Image	Trace Cutting Image
ESP32		
ESP8266		

3.2.2.2. Radiation Test

- Use the PCB antenna directly and run the test using a shielded box.
- Make sure to configure the correct path attenuation for a specific radiation test.

3.2.3. Pin Wiring Diagram



ESP32 / 8266 Products:

Download Mode: EN (pulled high), GPIO15 (pulled low, only for ESP8266), GPIO0 (pulled low)

Flash Operation Mode: EN (pulled high), GPIO15 (pulled low, only for ESP8266), GPIO0 (pulled high or floating)

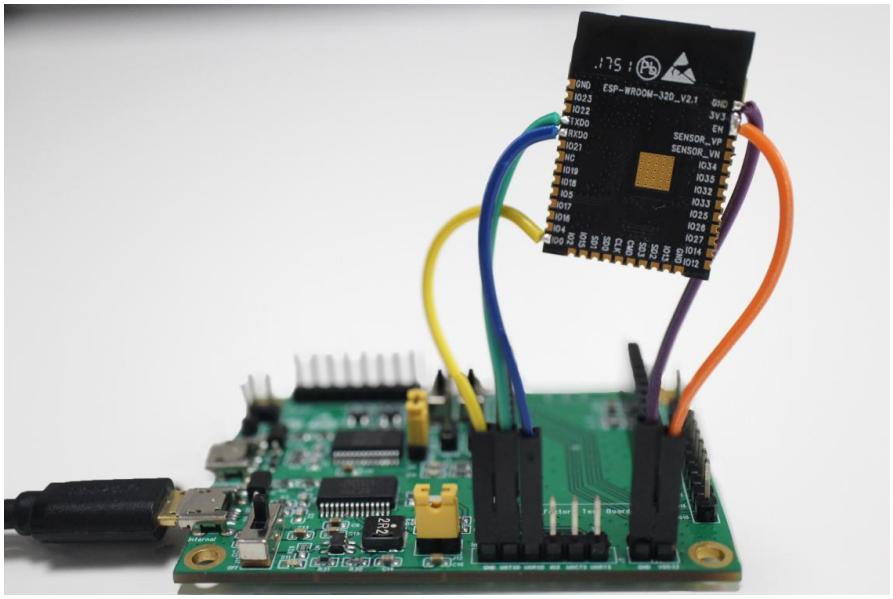


Figure 3-3. Test Scenario 1 - Wiring for the DUT as a standalone controller

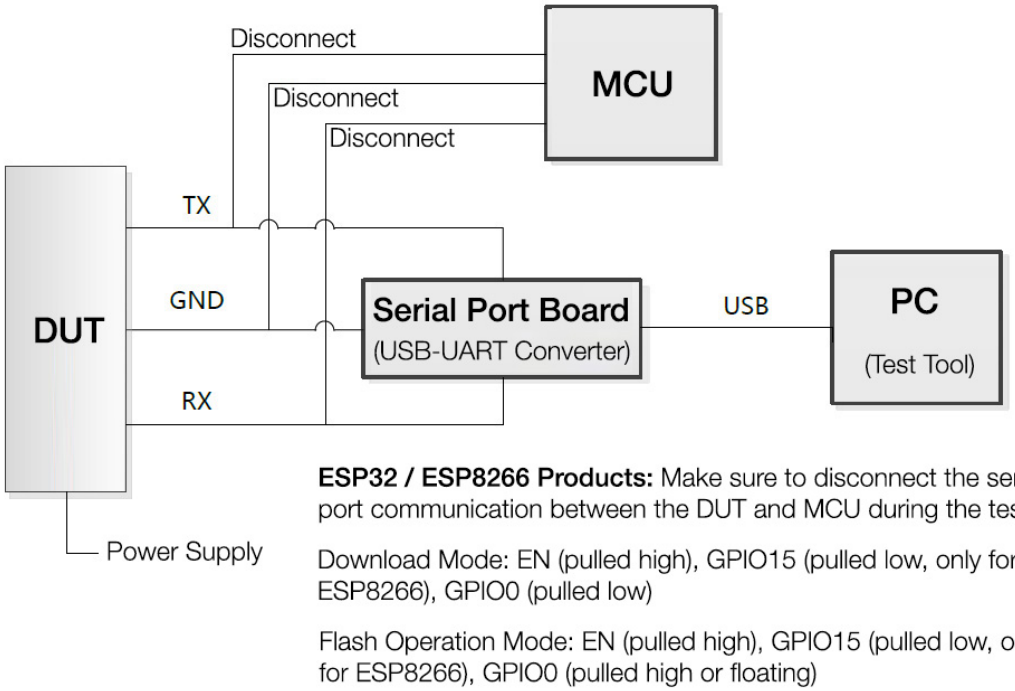


Figure 3-4. Test Scenario 2 - Wiring for the DUT as an integrated part of MCU

📖 ***Note:***

- *The serial port board is connected to the PC with a micro-USB cable. For the driver installation, please refer to Appendix A.*
- *For the test scenario 2, make sure to disconnect the serial port communication between the DUT and MCU; otherwise, the command sending process from the PC to DUT will be interfered with.*

3.3. Software Installation

- Install the tester-related software (e.g. the softwares related to IQView).
- Install EspRFtestTool, and run the tool (make sure all the connected serial ports function properly).

4. RF Test Tool

4.1. Features

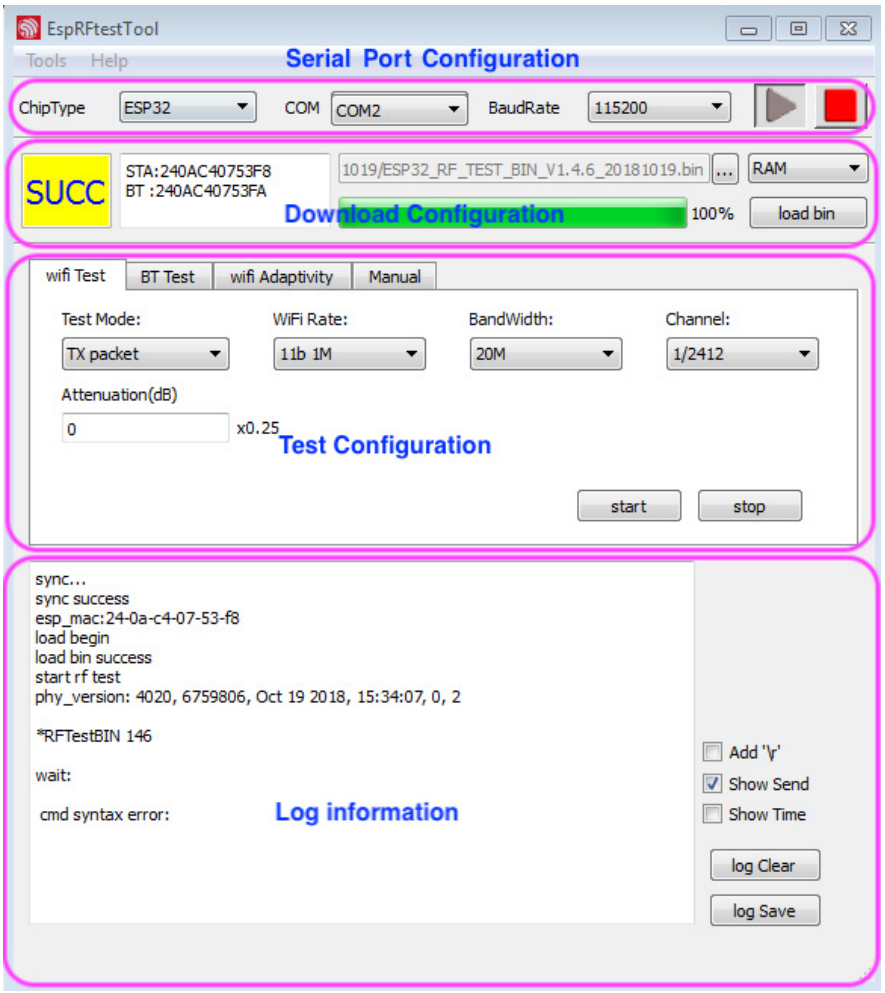


Figure 4-1. EspRFtestTool Interface

As shown in figure 4-1, EspRFtestTool has the following main features:

- Supports test firmware downloading to RAM or flash.
- Offers four RF tests: Wi-Fi, BT, Wi-Fi Adaptivity, and manual command input test.
- Printing and saving log information.
- Online help documentation.

4.2. User Interface

The user interface of the RF test tool has four sections, described in Table 4-1.

Table 4-1. Tool Overveiw

Section	Description
Serial Port Configuration	The configuration options include chip type, COM port, baud rate, and the status of the serial port.
Download Configuration	The configuration items include a bin-browse button (the one with tree dots), download location (RAM or flash), load bin (download button), and download status showing “SYNC”, “LOAD”, “SUCC”, or “FAIL”.
Test Configuration	The test items include Wi-Fi, BT, Wi-Fi Adaptivity and Manual (manual command input test). Click on ***start*** to begin a test, and ***stop*** to terminate a test.
Log Information	All the operating data is printed in this section. You can save or clear the log output.

📖 ***Note:***

- *This guide demonstrates the RF test tool with the ESP32 product as an example.*
- *Make sure the correct chip type and bin files are selected while testing ESP8266 products.*

5. RF Performance Test

5.1. Test Demonstration

5.1.1. Configure the Serial Port

Double-click ***espRFTool.exe*** to run the EspRFtestTool:

- Choose ***ESP32*** for **ChipType**
- Choose the correct port number for ***COM*** (***COM27*** in this case)
- Set the ***BaudRate*** to ***115200***
- Switch the serial port status to *open* (the port can be switched between the *open* and closed statuses)

5.1.2. Download the Firmware

- Browse to the path where test bin files are saved, for example: **D:/ESP32_TEST/** ***ESP32_RF_TEST_BIN_XXXX.bin.***
- Choose the download location: *RAM* or *flash* (for details on their difference, please refer to Section 2.2).
- MAC address: The status block shows the MAC address of the chip during the download process.
- Click on ***load bin*** and start the download process.
- ***SUCC*** will appear if the downloading was successful.

> ⚠ ***Notice:***
>
> - *If the bin file is downloaded to flash, you need to download it only once. To run the downloaded file, you need to switch to Flash operation mode by floating GPIO0 and re-powering the DUT.*
> - *If the bin file is downloaded to RAM, it will run immediately after the download process is completed. But the RAM program will be erased as soon as you reboot the DUT. You will have to download the bin file again.*

5.1.3. Run the RF test

5.1.3.1. For Wi-Fi test, select *wifi Test*

1. The following ***Test Mode*** options are available for ***wifi Test***:

   TX continues: continuous TX mode

   TX packet: non-continuous TX mode

   RX packet: RX mode

   TX tone: single carrier TX mode

   Choose the ***Test Mode*** according to your specific requirements.

2. To test TX, choose ***TX packet*** for **Test Mode**, and set ***WiFi Rate***, ***BandWidth***, ***Channel***, and ***Attenuation***. TX data rates are listed in Table 5-1.

Table 5-1. TX Data Rates

11b		11g		11n	
Parameter	Data rate	Parameter	Data rate	Parameter	Data rate
0x0	1 Mbps	0xb	6 Mbps	0x10	6.5 Mbps/MCS0
0x1	2 Mbps	0xf	9 Mbps	0x11	13 Mbps/MCS1
0x2	5.5 Mbps	0xa	12 Mbps	0x12	19.5 Mbps/MCS2
0x3	11 Mbps	0xe	18 Mbps	0x13	26 Mbps/MCS3
-	-	0x9	24 Mbps	0x14	39 Mbps/MCS4
-	-	0xd	36 Mbps	0x15	52 Mbps/MCS5
-	-	0x8	48 Mbps	0x16	58.5 Mbps/MCS6
-	-	0xc	54 Mbps	0x17	65 Mbps/MCS7

3. For example, if the ***WiFi Rate*** is set to ***MCS7***, ***Bandwidth*** to ***20 M***, and ***Channel*** to ***1***, you may choose the options as shown in Figure 5-1, and click on ***start*** to begin the test. You will get the log output described below:

   - *tx_contin_en 1*: continuous TX mode with a duty ratio of 92%, mainly used for the certification test
   - *tx_contin_en 0*: TX packet mode, mainly used for the performance test
   - *tx_cbw40m_en 0*: 11n HT20, 20 M bandwidth
   - *tx_cbw40m_en 1*: 11n HT40, 40 M bandwidth (supported by ESP32 only)

> ⚠ ***Notice:***
>
> - *Before switching to other channels or data rates, please click on **stop** to terminate the current transmission.*
> - *ESP32 supports HT40 and HT20, while ESP8266 supports HT20 only.*
> - *HT40 and HT20 use the identical way to transmit packets.*

4. Click on ***stop*** to end the TX process, the returned log message will be *cmdstop*.

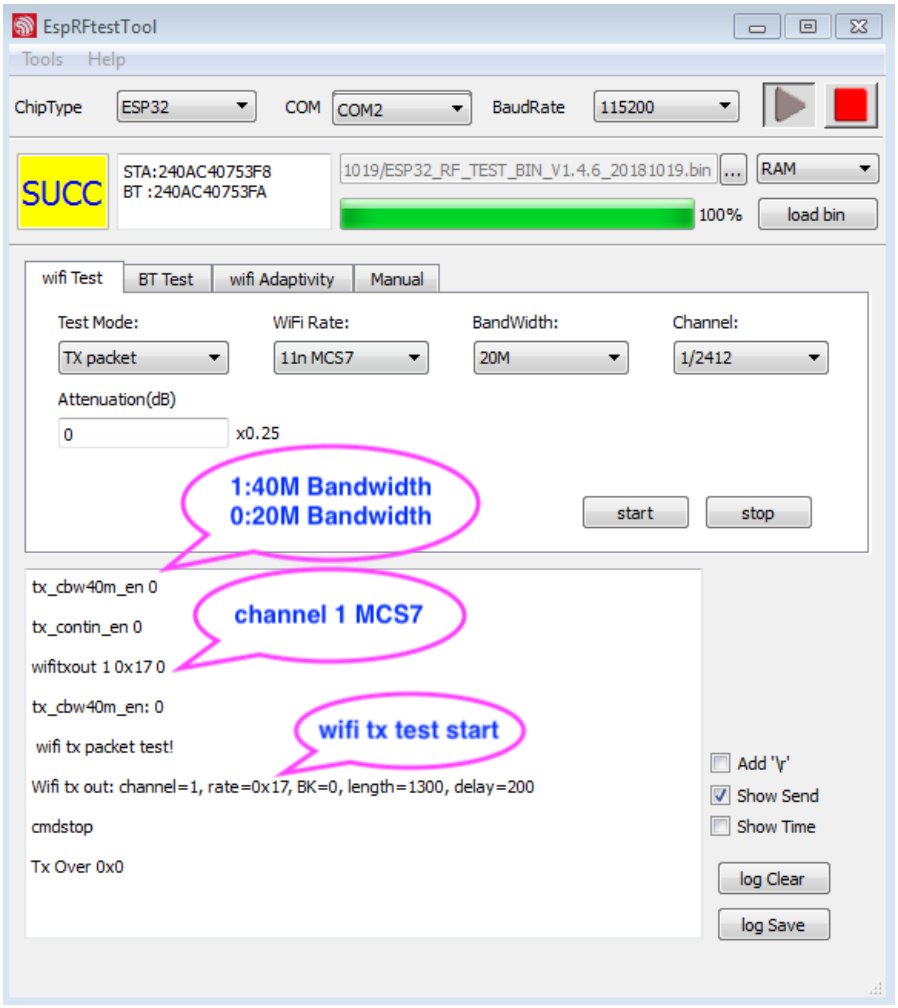


Figure 5-1. Log Output of the ESP32 Wi-Fi TX Test

5. Wi-Fi RX performance test:
   - Choose ***RX packet*** for ***Test Mode***, and set ***WiFi Rate***, ***BandWidth*** and ***Channel*** according to your requirements. Click on ***start*** to begin the test.
   - After RX is done, click on ***stop*** to end the test.
     - *Correct* indicates the number of the received packets in total.
     - *Desired* indicates the number of the received packets at a specifically set data rate.
     - *RSSI* indicates the average power of the received Desired packets.

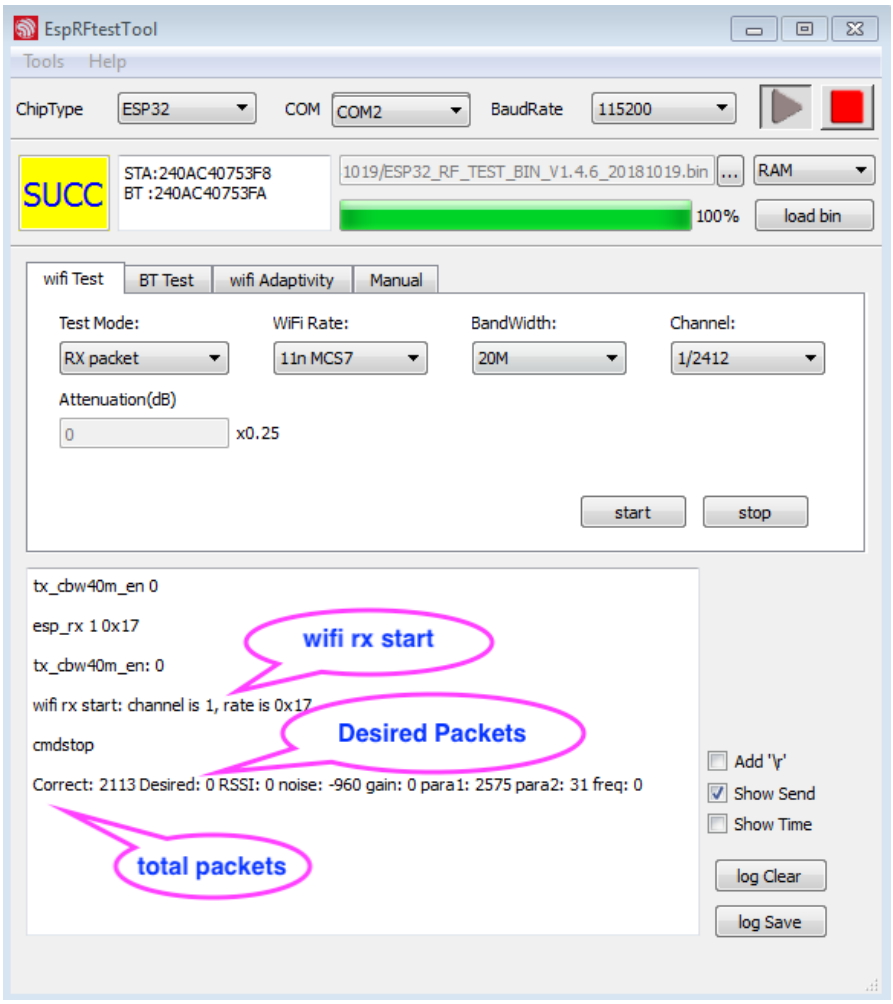


Figure 5-2. Log Output of the ESP32 Wi-Fi RX Test

5.1.3.2. For BT test, select *BT Test*

1. To test the BT TX performance with the data rate BR 1M, the DH type DH1, the data type 1010, and the TX channel 0, please refer to Figure 5-3 for your configurations. The log information section gives you the TX details.
2. Log information: Take the BT TX test as an example, *fcc_bt_tx* is followed by seven parameters described below:
   - <Parameter1> TX power level with a range of 0-7 (typically 4), in multiples of 3 dB.
   - <Parameter2> Enables or disables frequency hopping: 1 - enable; 0 - disable.
   - <Parameter3> Selects the TX channel from 0 to 78.
   - <Parameter4> Selects the frequency modulation type: 1 - 1M; 2 - 2M; 3 - 3M.
   - <Parameter5> Selects the DH type: 1 - DH1; 3 - DH3; 5 - DH5.
   - <Parameter6> Selects the data type: 0 - 1010; 1 - 00001111; 2 - prbs9.
   - <Parameter7> Reserved: 0 (optional in the command).

> ⚠ ***Notice:***
>
> - *ESP8266 only supports Wi-Fi.*
> - *ESP32 supports Wi-Fi+BT, so BT test is only for ESP32.*

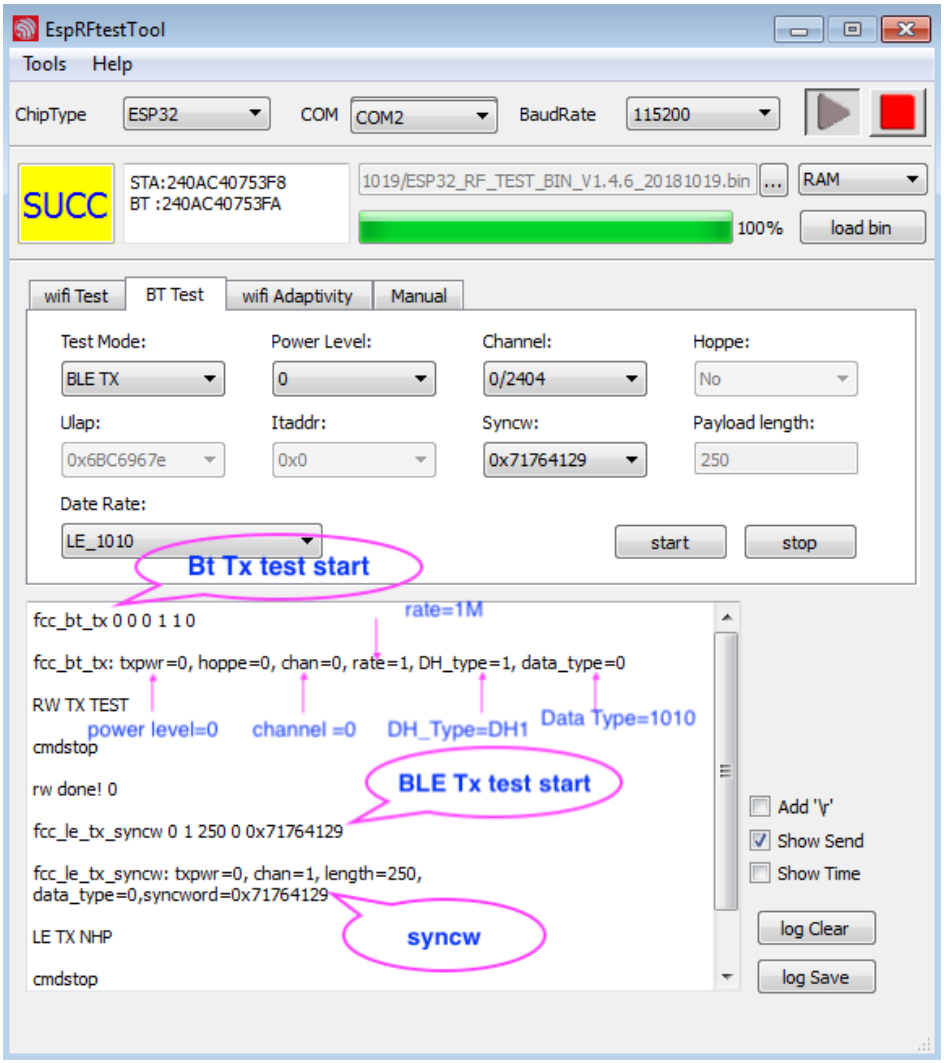


Figure 5-3. Log Output of the BT TX Test

> ⚠ ***Notice:***
>
> - *For the test of BLE TX, make sure "0x71764129" is selected for **Syncw**.*

3. BT RX performance test

   (1) To test the RX performance of LE channel 0, choose ***BLE RX*** for ***Test Mode***, ***0/2404*** for ***Channel***, and ***LE_prbs9*** for ***Data Rate***. Please refer to Figure 5-4.

   (2) When RX is done, click on ***stop*** to check the detailed information of the received packets. The format of the number of received packets is: 0 0 0 0 0 0 0 0 w 0 0 0 0 0 0 0 p 0 0 0 b 0 0

- The first parameter in hexadecimal format indicates the total number of the received packets.
- The second parameter in hexadecimal format indicates the number of the received packets at a specific data rate.
- The last parameter in hexadecimal format indicates the number of bit errors.
- The second-to-last hexadecimal parameter indicates the total number of the received bits at a specific data rate.

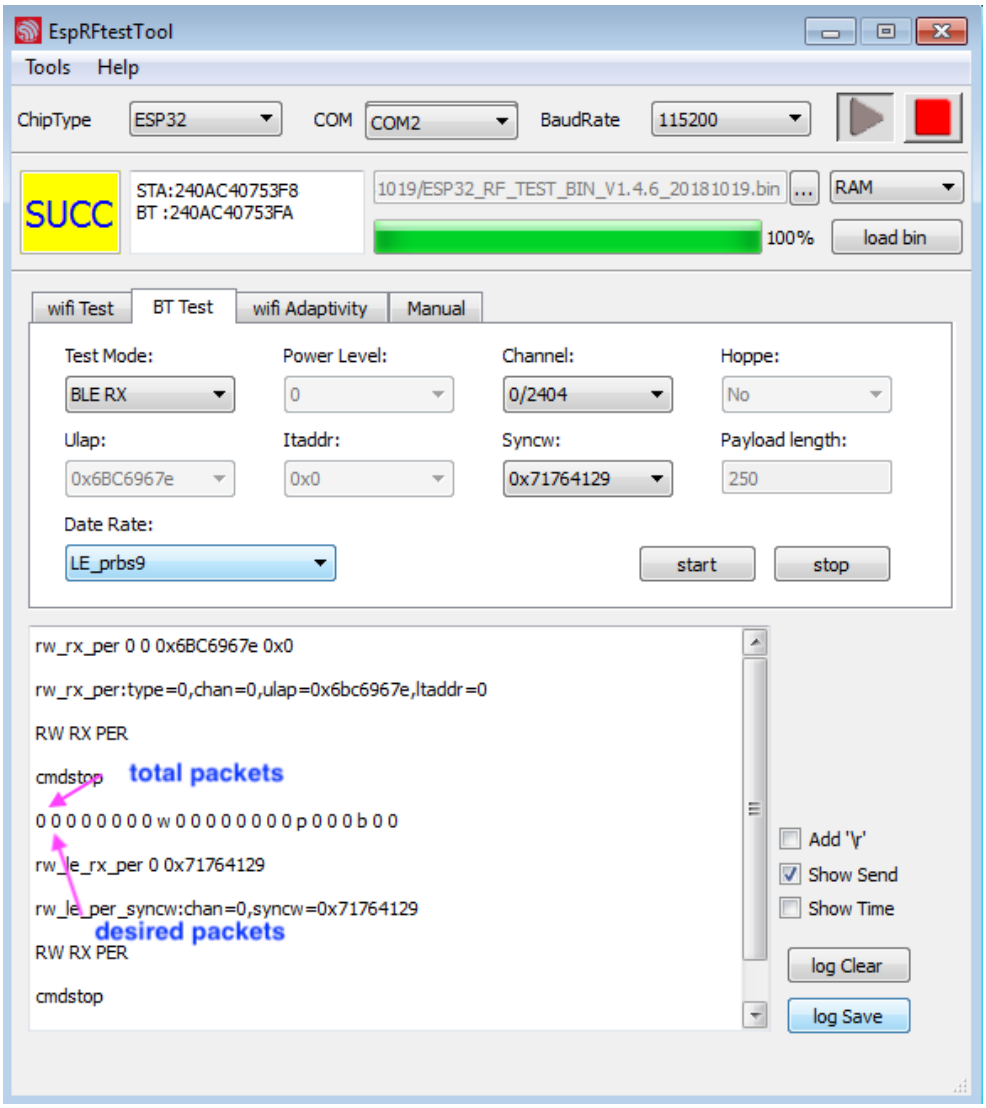


Figure 5-4. Log Output of the BT RX Test

> 📖 ***Note:***
>
> - *Choose **Manual** to manually input commands. Ensure to check **Add '\r'** before you **send** commands.*
> - *In case of a wireless tester IQView is used to test the DUT's RF performance on channel 1 at a data rate of 11n MCS7, the test results will look like the ones in Figure 5-5.*

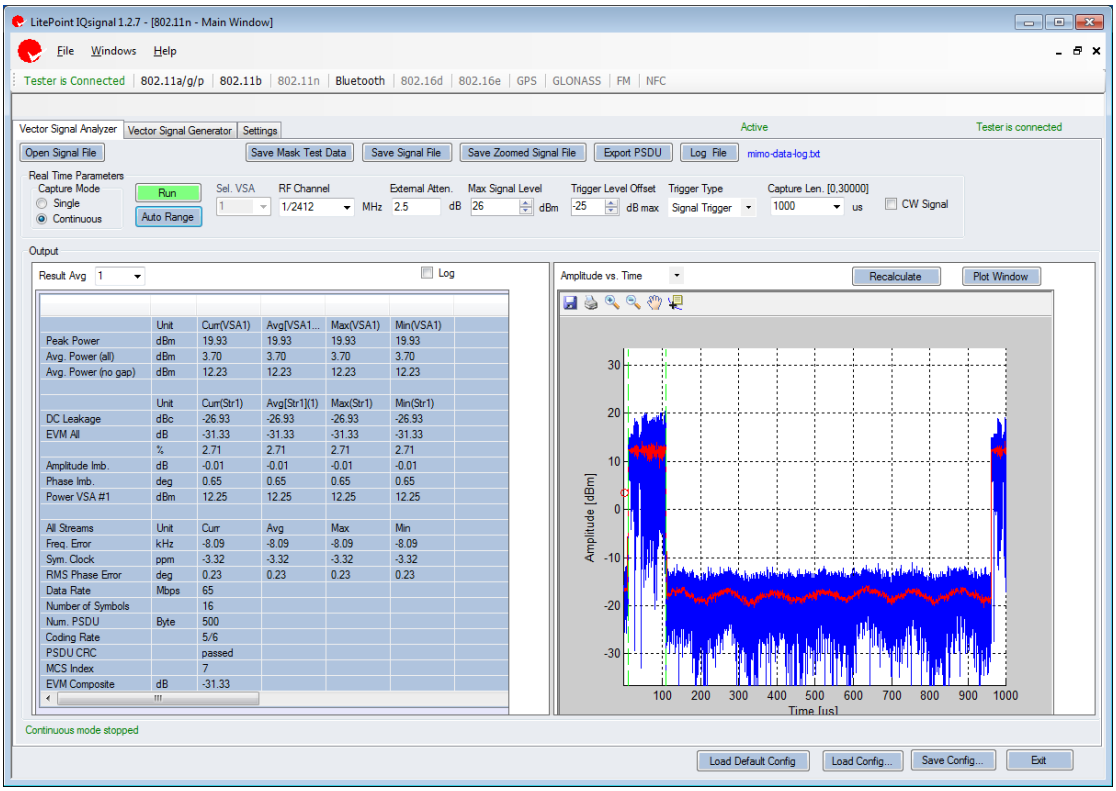


Figure 5-5. RF Performance Test by IQView

> 📖 ***Note:***
>
> *You need to set a correct value for the path loss; otherwise, the test results will be inaccurate.*

5.2. Online Help Documentation

Click ***Help*** on the menu bar to access the most recent online help documentation. You may also visit the link: *http://espressif.com/en/support/download/other-tools*.

6. RF Certification Test

6.1. Wi-Fi Certification Test

6.1.1. Environment Setup

1. Download the firmware ***ESP32_RF_TEST_BIN_XXXX.bin*** to flash and choose ***115200*** for ***BaudRate***.
2. When the download process is completed, disconnect GPIO0 and re-power the DUT.
3. Run the bin files. When the setup is done, the DUT is ready for the configuration with the RF test tool to start the test.

> 📖 ***Note:***
>
> *Follow the instructions in Chapter 5 to configure the serial port and download the firmware.*

6.1.2. Test-related Configuration

1. Select ***wifi Test***, and choose ***TX continue*** for ***Test Mode***. For ***WiFi Rate***, ***BandWidth***, ***Channel*** and ***Attenuation***, choose the right options according to the certification test requirements. See the interface below:

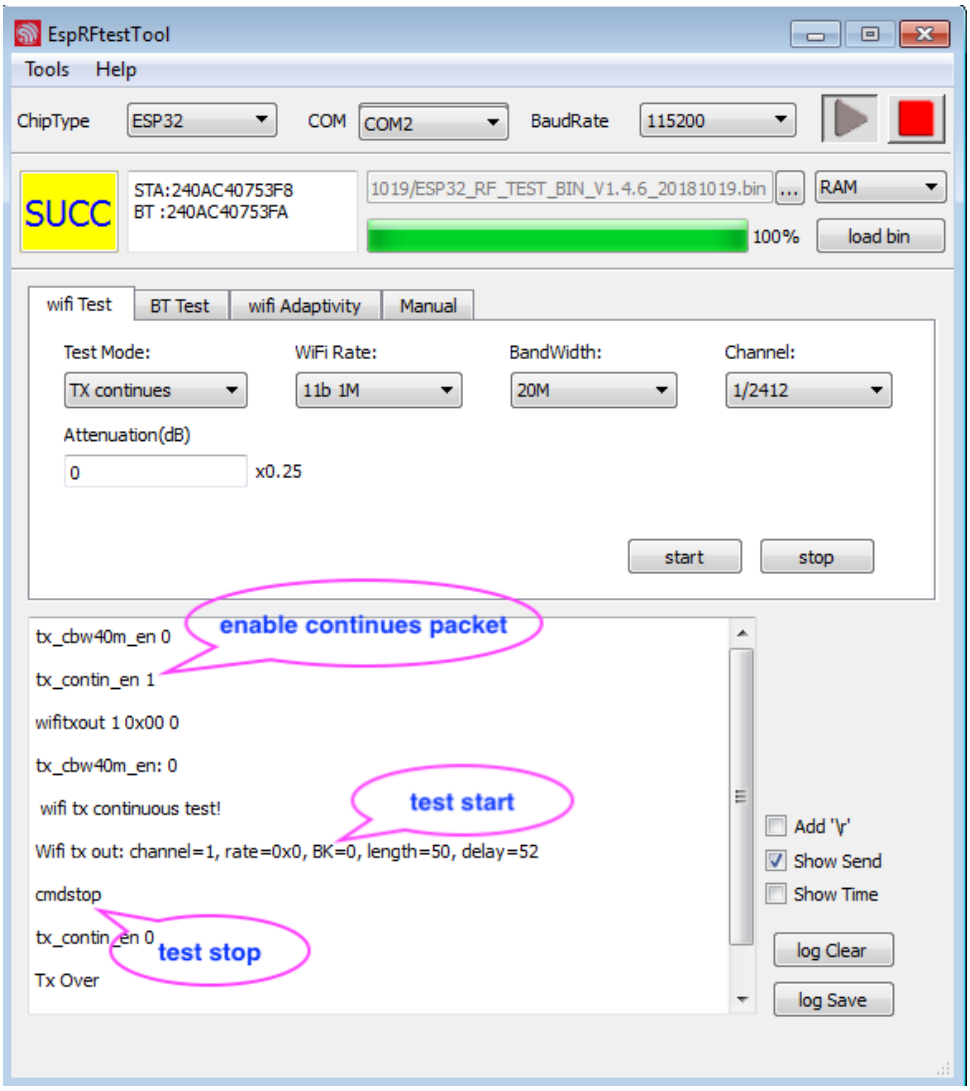


Figure 6-1. Log Output of the WiFi TX Test

> 📖 ***Note:***
>
> 1. *About the options under **Test Mode**:*
>
> - ***TX continue** is mainly for a certification test, and it provides a TX duty cycle of 92%;*
> - ***Wifiscwout** is mainly for single carrier TX.*
>
> 2. ***Attenuation** indicates the power attenuation in multiples of 0.25 dB. You may set the power attenuation according to the certification requirements.*

2. Click on ***start*** to begin sending commands. TX data rates are listed in Table 5-1 for your reference.
3. Click on ***stop*** to terminate the command sending process.

> 📖 ***Note:***
>
> *During the command sending process, make sure to click on **stop** before switching to other channels or TX data rates.*

6.2. BT Certification Test

6.2.1. Environment Setup

1. Download the firmware ***ESP32_RF_TEST_BIN_XXXX.bin*** to flash and choose ***115200*** for ***BaudRate***.
2. When the download process is completed, disconnect GPIO0 and re-power the DUT.
3. Run the bin files. When the setup is done, the DUT is ready for configuration with the RF test tool to start the test.

6.2.2. Test-related Configuration

The BT certification test shares the same testing process as the BT test that is described in Chapter 5.

7. Adaptivity Certification Test

7.1. Environment Setup

1. Download the firmware:
   - For ESP32: download ***ESP32_Adaptivity_XXXX.bin*** to flash and choose ***115200*** for ***BaudRate***.
   - For ESP8266: download ***ESP8266_Adaptivity_XXXX.bin*** to flash and choose ***74880*** for ***BaudRate***.
2. When the download process is completed, disconnect GPIO0 and re-power the DUT.
3. Run the bin files. When the setup is done, the DUT is ready for configuration with the RF test tool to start the test.

7.2. STA Mode

- Select ***wifi Adaptivity***. For the product in Station mode, the interface is shown in Figure 7-1 below.

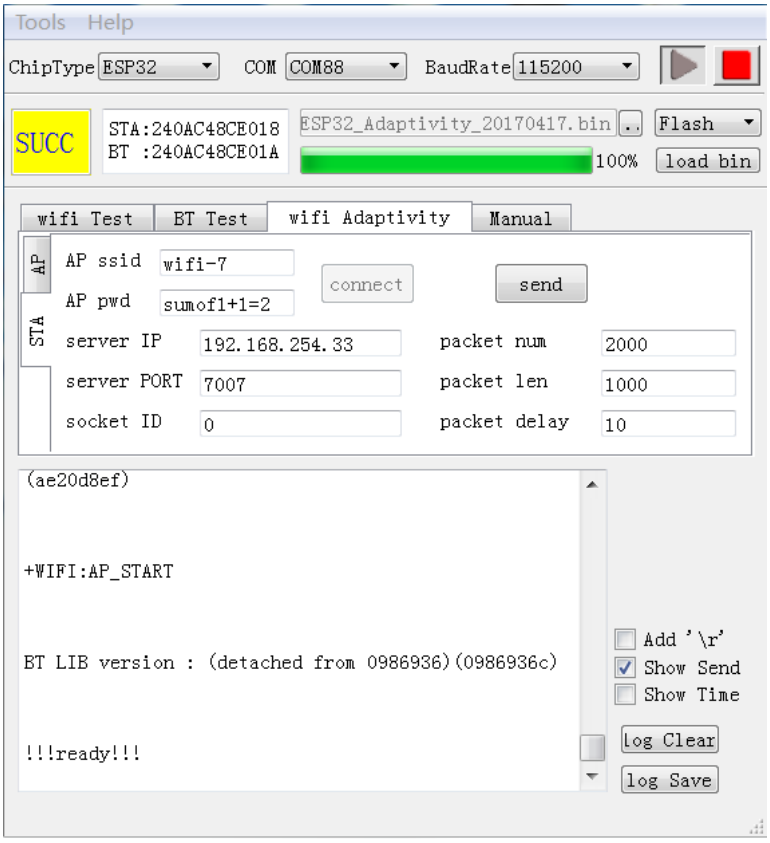


Figure 7-1. Adaptivity Test in STA Mode

- For STA mode, the parameters are described in the table below:

Table 7-1. Description of the STA-mode Parameters

Parameters	Description
AP ssid	Name of the target AP (router)
AP pwd	Password for the target AP (router)
Server IP	IP of the target host to be connected
Server PORT	Port of the target host to be connected
Socket ID	UDP socket of the DUT (0 by default)
Packet num	Number of TX UDP packets
Packet len	Length of TX UDP packets
Packet delay	TX interval of UDP packets (in ms)

- Test steps:
  - Fill in the ***AP ssid*** and ***AP pwd*** fields, and then click on ***connect*** to connect to AP.
  - After the connection to AP is established, set TX parameters (see Table 7-1), and click on ***send*** to start TX.

> ⚠ ***Notice:***
>
> *If starting TX takes a while, set a larger value for **Packet num**, for example, **6000000**.*

7.3. AP Mode

- Select ***wifi Adaptivity*** and choose ***AP*** mode.

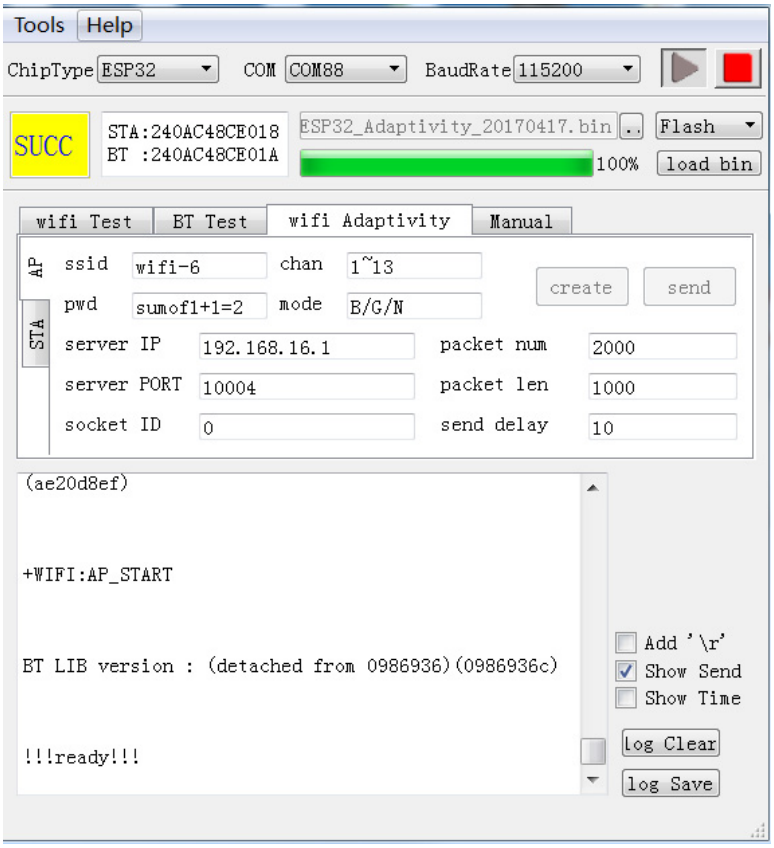


Figure 7-2. Adaptivity Test in AP Mode

- For AP mode, the parameters are described in the table below:

Table 7-2. Description of the AP-mode Parameters

Parameters	Description
ssid	Name of the DUT as AP
pwd	Password for the DUT as AP
chan	Channel of the DUT as AP, ranging from 1 to 13
mode	Mode of the DUT as AP, for example, B/G/N
Server IP	IP of the target host to be connected
Server PORT	Port of the target host to be connected
Socket ID	UDP socket of the DUT (0 by default)
Packet num	Number of TX UDP packets
Packet len	Length of TX UDP packets
send delay	TX interval of UDP packets (in ms)

Test steps:

- Fill in the ***ssid*** and ***pwd*** fields, and then click on ***create*** to create AP (hotspot).
- When AP is successfully created, it is ready for the connection initiated by other stations.
- Once the connection is established, set TX parameters, and click on ***send*** to start TX.

> ⚠ ***Notice:***
>
> *If starting TX takes a while, set a larger value for **Packet num**, for example, **6000000**.*

8. ESP32 Wi-Fi/BT Operation Command

To test the EMC/RF performance, please download *ESP32_RF_TEST_BIN_V1.4.6_20181019.bin*. Table 8-1 shows the download addresses.

Table 8-1. Download Addresses for the Binaries

Binaries	Download Address
ESP32_RF_TEST_BIN_V1.4.6_20181019.bin	0x1000

8.1. Wi-Fi Test Commands

> 📖 ***Note:***
>
> *The baud rate is 115200.*

8.1.1. Continuous Tx Enable Command

`tx_contin_en <Parameter1>`

`<Parameter1>:` Certification test 100% duty cycle Tx enable, 1 enable 100% duty cycle Tx; 0 off, default mode, for instrument testing such as iqview.

Command: tx_contin_en 1

Return: wifi tx continuous test!

Command: tx_contin_en 0

Return: wifi tx packet test!

8.1.2. Tx Start Command

`wifitxout <Parameter1> <Parameter2> <Parameter3>`

`<Parameter1>`: Select the Tx channel from 1 ~ 14.

`<Parameter2>`: Select the Tx data rate according to Table 8-2.

`<Parameter3>`: Tx power attenuation. It is an 8-bit signed operand, in multiples of 0.25 dB. For example, 4 indicates an attenuation of 1 dB.

Table 8-2. Parameter2 and Tx Data Rate

11b		11g		11n	
Parameter	Data rate	Parameter	Data rate	Parameter	Data rate
0x0	1 Mbps	0xb	6 Mbps	0x10	6.5 Mbps / MCS0
0x1	2 Mbps	0xf	9 Mbps	0x11	13 Mbps / MCS1
0x2	5.5 Mbps	0xa	12 Mbps	0x12	19.5 Mbps / MCS2
0x3	11 Mbps	0xe	18 Mbps	0x13	26 Mbps / MCS3
-	-	0x9	24 Mbps	0x14	39 Mbps / MCS4
-	-	0xd	36 Mbps	0x15	52 Mbps / MCS5
-	-	0x8	48 Mbps	0x16	58.5 Mbps / MCS6
-	-	0xc	54 Mbps	0x17	65 Mbps / MCS7

> ⚠ ***Notice:***
> *After sending data packets, please input the Stop command to put an end to the process.*

Example:

```
wifitxout  1  0x0  0
```

Print: `Wifi tx out: channel=1, rate=0x0, BK=0`

Send the packets through channel 1 (2412 MHz) with a data rate of 1 Mbps.

8.1.3. Tx Stop Command

```
cmdstop
```

Print: `Tx Over`

> ⚠ ***Notice:***
> *After sending data packets, please input the Stop command to change the channel or data rate.*

8.1.4. Command for Selecting Tx/Rx 11n Mode 20 Mbps or 40 Mbps

```
cbw40m_en  <Parameter1>
```

- Command for selecting the 11n HT20 mode:

  ```
  cbw40m_en  0
  ```

  Print: `cbw40m_en: 0`

- Command for selecting the 11n HT40 mode:

  ```
  cbw40m_en  1
  ```

  Print: `cbw40m_en: 1`

> 📖 ***Note:***
>
> *The 20 Mbps and 40 Mbps modes share the same command.*

8.1.5. Rx Start Command

`esp_rx <Parameter1> <Parameter2>`

`<Parameter1>`: Select the Rx channel from 1 ~ 14.

`<Parameter2>`: Select the Rx data rate according to Table 8-2.

> ⚠ ***Notice:***
>
> *After receiving data packets, please input Stop command to put an end to the process.*

Example:

`esp_rx 1 0x0`

Print: `wifi rx start: channel is 1, rate is 0x0`

Receive the packets in channel 1 (2412 MHz) with a data rate of 1 Mbps.

8.1.6. Rx Stop Command

`cmdstop`

Print: `Correct: 0 Desired: 0 RSSI: 0`

`Correct` indicates the number of received packets.

`Desired` indicates the number of received packets with the corresponding data rate of `<Parameter2>`.

`RSSI` indicates the average power of the `Desired` packets received.

8.1.7. SCW Tx Command

`wifiscwout <Parameter1> <Parameter2> <Parameter3>`

`<Parameter1>`: SCW Tx enable signal, 1 = send; 0 = stop.

`<Parameter2>`: Select the SCW Tx channel from 1 ~ 14.

`<Parameter3>`: SCW power attenuation. The unit is 0.25 dB. For example, 4 indicates an attenuation of 1 dB and 8 indicates an attenuation of 2 dB.

- SCW Tx example:

  `wifiscwout 1 14 0`

  Print: `wifi single carrier tx out`

  Transmit SCW in channel 14 (2484 MHz)

- SCW Tx stop example:

  `wifiscwout 0 14 0`

  Print: `wifi single carrier tx stop`

Stop SCW transmission.

8.2. BT Test Commands

> 📖 ***Note:***
>
> *Baud rate is 115200.*

8.2.1. BR/EDR Tx Command

`fcc_bt_tx <Parameter1> <Parameter2> <Parameter3> <Parameter4> <Parameter5> <Parameter6> <Parameter7>`

`<Parameter1>`: Tx power attenuation. The range is 0 ~ 7. The unit is 3 dB; Normally the value is 4 and the power is about 0 dBm.

`<Parameter2>`: Enable or disable frequency hopping. 1: enable; 0: disable.

`<Parameter3>`: Select the Tx channel from 0 ~ 78.

`<Parameter4>`: Select the modulation mode. 1: 1M; 2: 2M; 3: 3M.

`<Parameter5>`: Select the DH type. 1: DH1; 3: DH3; 5: DH5.

`<Parameter6>`: Select the Data type. 0: 1010; 1: 00001111; 2: prbs9.

`<Parameter7>`: Reserved: 0 (optional in the command).

Example:

```
fcc_bt_tx  4  0  0  1  3  1  0
```

Print:

```
fcc_bt_tx: txpwr=4, hoppe=0, chan=0, rate=1, DH_type=3, data_type=1
RW TX TEST
```

The command indicates that the Tx power attenuation level is 4; frequency hopping is disabled; the channel is 0 (2402 MHz); the data rate is BR1M; the DH type is DH3 and the data type is 00001111.

8.2.2. LE Tx Command

Package identification is not included:

`fcc_le_tx <Parameter1> <Parameter2> <Parameter3> <Parameter4> <Parameter5>`

`<Parameter1>`: Tx power attenuation. The range is 0 ~ 7. The unit is 3 dB; Normally the value is 4 and the power is about 0 dBm.

`<Parameter2>`: Select the Tx channel from 0 ~ 39.

`<Parameter3>`: Select the payload length. The range is 0 ~ 255. The unit is byte. Normally the value is 250.

`<Parameter4>`: Select the Data type. 0: 1010; 1: 00001111; 2: prbs9.

<Parameter5>: Reserved: 0 (optional in the command).

Example:

```
fcc_le_tx  4  0  250  2  0
```

Print:

```
fcc_le_tx: txpwr=4, chan=0, length=250, data_type=2
RW LE TX NHP
```

The command indicates that the Tx power attenuation level is 4; the channel is 0 (2402 MHz); the data rate is LE1M and the data type is prbs9.

Package identification is included:

fcc_le_tx_syncw <Parameter1> <Parameter2> <Parameter3> <Parameter4> <Parameter5>

<Parameter1>: Tx power attenuation. The range is 0 ~ 7. The unit is 3 dB; Normally the value is 4 and the power is about 0 dBm.

<Parameter2>: Select the Tx channel from 0 ~ 39.

<Parameter3>: Select the payload length. The range is 0 ~ 255. The unit is byte. Normally the value is 250.

<Parameter4>: Select the Data type. 0: 1010; 1: 00001111; 2: prbs9.

<Parameter5>: The package's identity. syncw=0x71764129 bu default.

Example:

fcc_le_tx_syncw 4 0 250 2 0x71764129

Print:

fcc_le_tx_syncw: txpwr=4, chan=0, length=250, data_type=2,syncword=0x71764129

RW LE TX NHP

The command indicates that the Tx power attenuation level is 4; the channel is 0 (2402 MHz); the data rate is LE1M and the data type is prbs9. The package's identity syncw is 0x71764129.

8.2.3. Tx Stop Command

```
cmdstop
```

8.2.4. BR/EDR Rx Start Command

rw_rx_per <Parameter1> <Parameter2> <Parameter3> <Parameter4> <Parameter5>

<Parameter1>: 0: BR; 1: EDR.

<Parameter2>: Select the Rx channel from 0 ~ 78. 0 to 39 represent even-numbered channels, and 40 to 78 represent odd-numbered channels. For example, if Parameter2 is 0, channel 0 is selected; if Parameter2 is 1, channel 2 is selected; if Parameter2 is 2, channel 4 is selected, and so on. So if Parameter2 is 39, channel 78 is selected. In contrast, if Parameter2 is 40, channel 1 is selected; if Parameter2 is 41, channel 3 is selected; if Parameter2 is 42, channel 5 is selected, and so on. So if Parameter2 is 78, channel 77 is selected.

<Parameter3>: 32-bit Bluetooth address, including UAP (8-bit) and LAP (24-bit). The value is determined by the testing equipment.

<Parameter4>: The logical transport address specified by the protocol. The value is determined by the testing equipment with a range of 0 ~ 7.

<Parameter5>: Reserved: 0 (optional in the command).

Example:

```
rw_rx_per  1  40  0x6BC6967e  0  0
```

Print:

```
rw_rx_per:type=1, chan=40, ulap=0x6BC6967e, ltaddr=0
RW RX PER
```

The command indicates that the Rx data package is EDR (The DH type, Data type are selected by the equipment and can be modulated. DH1 is recommended.), the channel is 1 (2403 MHz), the data rate is 2M or 3M, the bluetooth address is 0x6BC6967e and the logical transport address is 0.

8.2.5. LE Rx Start Command

```
rw_le_rx_per <Parameter1>  <Parameter2>  <Parameter3>
```

<Parameter1>: Select the Rx channel number from 0 ~ 39. Channel 0 ,1, 2 ~ 10 correspond to the frequency of 2404 MHz, 2406 MHz, 2408 MHz ~ 2424 MHz, respectively. Channel 11, 12, 13 ~ 16 correspond to the frequency of 2428 MHz, 2430 MHz, 2432 MHz ~ 2478 MHz, respectively. Channel 37 corresponds to the frequency of 2402 MHz. Channel 38 corresponds to the frequency of 2426 MHz. Channel 39 corresponds to the frequency of 2480 MHz.

<Parameter2>: The data package identify (provided by the data generator or the equipment supplier).

<Parameter3>: Reserved: 0 (optional in the command).

Example:

```
rw_le_rx_per  11  0x71764129  0
```

Print:

```
rw_le_per_syncw: chan=11, syncw=0x71764129
RW RX PER
```

The command indicates that the Rx data package is LE (The Data type is usually prbs9.); the channel is 11 (2428 MHz); the data rate is 1M and the data identify is 0x71764129.

8.2.6. Rx Stop Command

`cmdstop`

Input `cmdstop` command to stop receiving packets. The serial port will print the number of packets that have been received.

The format of the number of received packets is
`0 0 0 0 0 0 0 0 w 0 0 0 0 0 0 0 0 p 0 0 0 b 0 0`

The first parameter (based on hexadecimal system) indicates the total number of packets received this time.

The second parameter (based on hexadecimal system) indicates the number of packets received of the corresponding speed at this time.

The last parameter (based on hexadecimal system) indicates the number of bit errors.

The second-to-last parameter (based on hexadecimal system) indicates the overall number of the received bits at a certain data rate.

8.2.7. BT Receiving Status Command

`cmdstatus`

Input `cmdstatus` command to print the receiving status. The serial port will print the number of packets that have been received.

The format of the number of received packets is
`0 0 0 0 0 0 0 0 w 0 0 0 0 0 0 0 0 p 0 0 0 b 0 0`

The first parameter (based on hexadecimal system) indicates the total number of packets received this time.

The second parameter (based on hexadecimal system) indicates the number of packets received of the corresponding speed at this time.

The last parameter (based on hexadecimal system) indicates the number of bit errors.

The second-to-last parameter (based on hexadecimal system) indicates the overall number of the received bits at a certain data rate.

8.2.8. SCW Tx Command

`bt_tx_tone <Parameter1> <Parameter2> <Parameter3>`

`<Parameter1>`: SCW Tx enable signal, 1 = send; 0 = stop.

`<Parameter2>`: Select the SCW Tx channel from 0 ~ 78.

`<Parameter3>`: SCW power attenuation. The unit is 0.25 dB. For example, 4 indicates an attenuation of 1 dB.

- SCW Tx example:

```
bt_tx_tone  1  0  0
```

Print: `BT TX TONE START!`

Transmit SCW in channel 14 (2402 MHz)

- SCW Tx stop example:

```
bt_tx_tone  0  0  0
```

Print: `BT TX TONE STOP!`

Stop SCW transmission.

9. ESP8266 Wi-Fi Operation Commands

To test the EMC/RF performance, please download the *ESP8266_RF_TEST_BIN_V127_26m_20180530.bin* file or the *ESP8266_RF_TEST_BIN_V127_40m_20180530.bin* file.

> 📖 ***Note:***
>
> *The case-sensitive commands should be input in the interactive window of* ***SecureCRT****. Please select* ***Chat Window*** *in the* ***View****.*

9.1. Test Commands

9.1.1. Continuous Tx Enable Command

`tx_contin_en <Parameter1>`

`<Parameter1>:` Certification test 92% duty cycle Tx enable, 1 enable 92% duty cycle Tx; 0 off, default mode, for instrument testing such as iqview.

Command: tx_contin_en 1

Return: wifi tx continuous test!

Command: tx_contin_en 0

Return: wifi tx packet test!

9.1.2. Tx Start Command

`wifitxout <Parameter1> <Parameter2> <Parameter3>`

`<Parameter1>`: Select the Tx channel from 1 ~ 14.

`<Parameter2>`: Select the Tx data rate according to Table 9-1.

`<Parameter3>`: It is an 8-bit signed operand that indicates Tx power attenuation. The unit is 0.25 dB. Value 4 means an attenuation of 1 dB, and 252 means a gain of 1 dB.

Table 9-1. <Parameter2> and Tx Data Rate

11b		11g		11n	
Parameter	Data rate	Parameter	Data rate	Parameter	Data rate
0x0	1 Mbps	0xb	6 Mbps	0x10	6.5 Mbps / MCS0
0x1	2 Mbps	0xf	9 Mbps	0x11	13 Mbps / MCS1
0x2	5.5 Mbps	0xa	12 Mbps	0x12	19.5 Mbps / MCS2
0x3	11 Mbps	0xe	18 Mbps	0x13	26 Mbps / MCS3
-	-	0x9	24 Mbps	0x14	39 Mbps / MCS4
-	-	0xd	36 Mbps	0x15	52 Mbps / MCS5
-	-	0x8	48 Mbps	0x16	58.5 Mbps / MCS6
-	-	0xc	54 Mbps	0x17	65 Mbps / MCS7

> ⚠ ***Notice:***
> *After sending data packets, please input the Stop command to end the process.*

Example:

```
wifitxout  1  0x0   0
```

Print: `Wifi tx out: channel is 1, data_rate is 11b 1.0Mb/s`

Send the packets via channel 1 (2412 MHz) with the data rate of 1 Mbps.

9.1.3. Tx Stop Command

```
cmdstop
```

Print: `Tx Over`

> ⚠ ***Notice:***
> *After sending data packets, please input the Stop command to switch the channel or data rate.*

9.1.4. Rx Start Command

```
esp_rx  <Parameter1>  <Parameter2>
```

`<Parameter1>`: Select the Rx channel from 1 ~ 14.

`<Parameter2>`: Select the Rx data rate according to Table 9-1.

> ⚠ ***Notice:***
> *After receiving data packets, please input Stop command to end the process.*

Example:

```
esp_rx  1  0x0
```

Print: `wifi rx start: channel is 1, rate is 0x0`

Receive the packets via channel 1 (2412 MHz) with a data rate of 1 Mbps.

9.1.5. Rx Stop Command

`cmdstop`

Print: `Correct: 0 Desired: 0 RSSI: 0`

`Correct` means the number of packets received.

`Desired` means the number of packets received with the corresponding data rate of <Parameter2>.

`RSSI` means the average power of the `Desired` packets received.

9.1.6. SCW Command

`wifiscwout <Parameter1> <Parameter2> <Parameter3>`

<Parameter1>: SCW sending enable signal, 1 = send; 0 = stop.

<Parameter2>: Select the SCW sending channel from 1 ~ 14.

<Parameter3>: It is an 8-bit signed operand that indicates SCW power attenuation. The unit is 0.25 dB. 4 means an attenuation of 1 dB, and 252 means a gain of 1 dB.

- SCW Tx example:

  `wifiscwout 1 14 0`

  Print: `wifi single carrier tx out`

  Transmit SCW in channel 14 (2484 MHz)

- SCW Tx stop example:

  `wifiscwout 0 14 0`

  Print: `wifi single carrier tx stop`

  Stop SCW transmission.

A. Appendix – Install a UART Driver on the Serial Port Board

1. Use a micro-USB cable to connect the serial port board to the PC. A UART driver will be installed automatically.

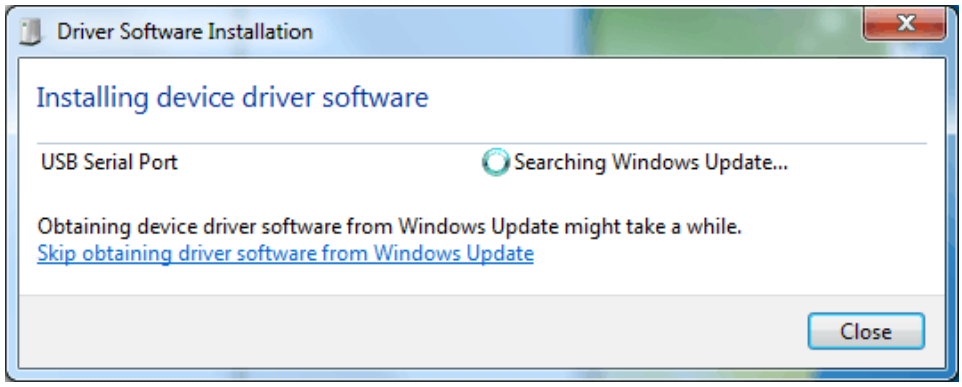


Figure A-1. Installing UART Driver on the PC

When the installation is finished, the following prompt will appear:

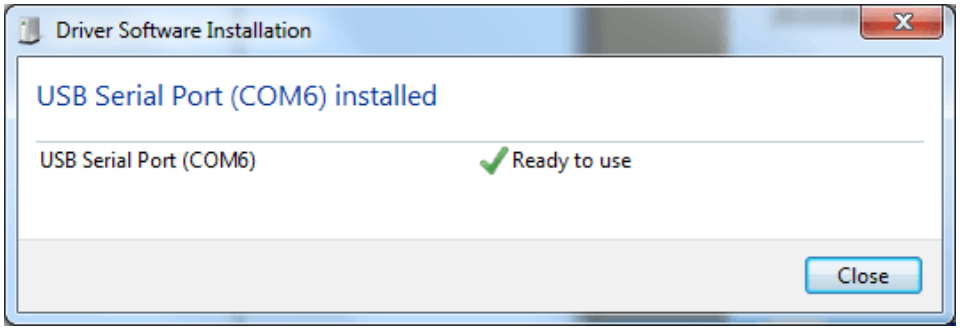


Figure A-2. UART Driver Installed

> ⚠ ***Notice:***
>
> *If the UART driver installation fails, you can install **./Tools/ft232r-usb-uart.zip** provided by Espressif.*

2. The RF test tool will automatically check the port number that will appear in the dropdown menu in the serial port configuration section.

Espressif IoT Team

www.espressif.com