ESP-Launcher Data Send and Receive Operation Power Consumption Tests

Version 1.0

About This Guide

The document is structured as follows:

Chapter	Title	Content
Chapter 1	Overview	Introduction to the tests.
Chapter 2	Organization of the Demonstration	Introduction to the architecture of the power consumption tests.
Chapter 3	Environment Setup	Introduction to the hardware and software setup of the power consumption tests.
Chapter 4	Test Procedure	Introduction to the procedures of the power consumption tests.
Chapter 5	Conclusion of the Demonstration	Conclusion of the power consumption tests.
Appendix I	Installing UART Driver of ESP-Launcher	Introduction to the installation of the UART driver on the PC, and how to use the UART debug tool.
Appendix II	Flash Download Tool Guide	Introduction to the Espressif flash download tool, and how to use it to download the firmware into the ESP-Launcher.
Appendix III	Python Script Guide	Introduction to the methods to run the Python script and to measure the electric current using ammeters.

Release Notes

Date	Version	Release Notes
2016.09	V1.0	Initial release.

Table of Contents

1. Overview

The power consumption of the ESP8266 depends on the PHY mode employed for sending or receiving data. In power critical applications, it is important to balance the average or burst data transfer rate as well as overall power consumption of the system.This test demonstration is intended to demonstrate the results of various power consumption tests on the ESP8266. Based on this guide, users can choose the most appropriate PHY mode and data transmission scheme for their application.

2. Organization of the Demonstration

In this test demonstration, the UART interface is used to send commands to the ESP8266 running the supplied firmware. The commands enable different test modes:

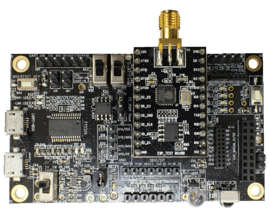
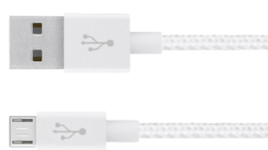
1. Packet send power operation consumption tests in different PHY modes can be categorized as follows:
   - Power consumption test when ESP-Launcher sends packets in the 802.11b mode, CCK 11 Mbps, Pout = +17 dB.
   - Power consumption test when ESP-Launcher sends packets in the 802.11g mode, OFDM 54 Mbps, Pout = +15 dB .
   - Power consumption test when ESP-Launcher sends packets in the 802.11n mode, MCS7, Pout = +13 dB.
2. Packet receive operation power consumption tests in different PHY modes can be categorized as follows:
   - Power consumption test on ESP-Launcher when using IEEE 802.11b standard, with 1024 bytes long packets received in an environment of -80 dB transmit power.
   - Power consumption test on ESP-Launcher when using IEEE 802.11g standard, with 1024 bytes long packets received in an environment of -70 dB transmit power.
   - Power consumption test on ESP-Launcher when using IEEE 802.11n standard, with 1024 bytes long packets received in an environment of -65 dB transmit power.

3. Environment Setup

3.1. Environment Preparation

3.1.1. Hardware Preparation

Table 3-1. Hardware Preparation

Name	Figure	Quantity	Description
ESP-Launcher (ESP8266 demo board)		1	The development board with ESP8266. Whip antenna is needed.
Micro USB Cable		1	To connect the ESP-Launcher to the PC. • PC provides power supply. • User can trace the log outputted from ESP8266 on the PC UART tool.
Ammeter		1	To measure the electric currents.
PC		1	PC for running demo tools: Windows XP or Windows 7 OS is recommended.
Router		1	To provide Wi-Fi network.

3.1.2. Software Preparation

Table 3-2. Software Preparation

Name	Location	Description
demo.bin	***./Bin-FW/***	The firmware running on the ESP-Launcher
SecureCRT.rar	***./Tools/***	PC UART terminal emulator tool (supports 74880 baud rate)
ft232r-usb-uart.zip	***./Tools/***	USB - UART converter driver
Python Script	***./Tools/***	Script of the electric current test (Python 2.7)
NI4882_1500f0.exe	***./Tools/***	Driver for the ammeter (GPIB interface)
Flash Download Tool	***./Tools/***	To download the firmware into the ESP-Launcher

3.2. Hardware Connection

- ESP-Launcher connects to the PC via the Micro USB cable. For instructions on how to install the UART driver on your PC, please refer to ***Appendix I***.
- The ammeter used in this demo connects to the PC through the GPIB interface.
- The ammeter should be connected into the circuit by serial lines, between the ESP8266 module and the 3.3V power supply. If you wish to measure the current consumption of your entire circuit, connect the ammeter in series with the main power source to your circuit.
- When testing the power consumption for receiving packets on the ESP-Launcher, place the ESP-Launcher in a network shield box to prevent external signal interference. Use a WT-200 Wireless Tester as the signal source to simulate a controlled situation to send corresponding signals to the module.

3.3. Software Installation

1. Flash Download Tool is for downloading the firmware into the ESP-Launcher.

> 📖 ***Notes:***
> - *The firmware to be downloaded is in **./Bin-FW/**.*
> - *For information how to use Flash Download Tool, please refer to **Appendix II**.*

2. For more on Python script to measure the current value, please refer to ***Appendix III.***

3.4. The ESP-Launcher

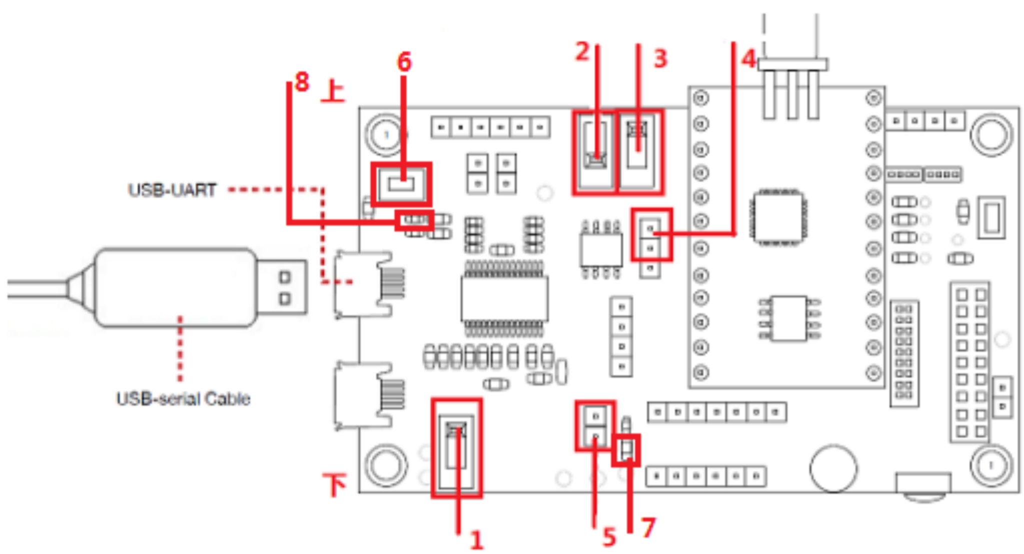


Figure 3-1. The ESP-Launcher

The switch “1”: toggle to the lower position

- Lower position: power-off
- Upper position: power-on

The switch “2”: toggle to the lower position

- Lower side: firmware download mode
- Upper side: program execution mode
- The ESP8266 will read the level of switch “2” when powering up.
  - If the switch "2" is set to the lower position, the ESP8266 will enter firmware download mode after power on.
  - If the switch "2" is set to the upper position, the ESP8266 will enter program execution mode after power on.

The switch “3” (CHIP_EN Pin.): toggle to the upper position

- Lower position: chip disable
- Upper position: chip enable

The pin “4”: put a jumper cap on the above 2 pins

The pin “5”: put a jumper cap on it

4. Test Procedure

4.1. Downloading Firmware

1. Download the supplied firmware to the ESP-Launcher. Use Flash Download Tool (ESP8266 DOWNLOAD TOOL) to download ***./Bin-FW/demo.bin*** into address ***0x0000***. For more details about using Flash Download Tool, please refer to ***Appendix II***.

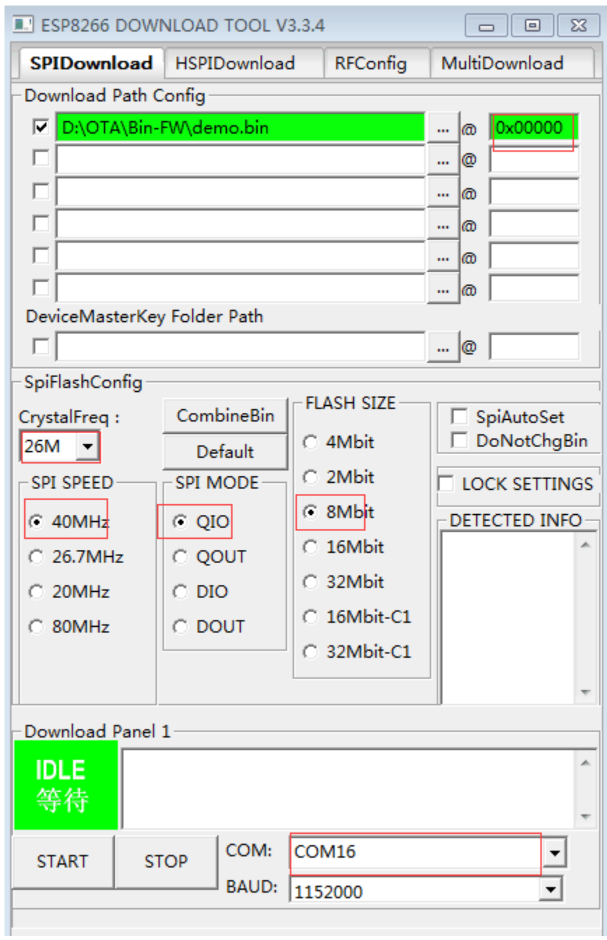


Figure 4-1. Downloading Firmware

> 📖 ***Notes:***
>
> - *In general, the firmware needs to be downloaded only once.*
> - *A new version of the firmware with improved functionality or new feature may be downloaded into the ESP-Launcher again.*

2. Run the firmware to confirm that the download is successful.
    - Connect the ESP-Launcher to the PC using the Micro USB cable.
    - Open the UART tool on your PC, and configure the baud rate to 115200.
    - Toggle the switch “2” on the ESP-Launcher to the upper position to set it into program execution mode.
    - Next, toggle the switch “1” to the upper position to power up the ESP-Launcher.
    - If the UART outputs the logs below, it means that the firmware is successfully downloaded.

Figure 4-2. UART Output Logs

4.2. Power Consumption Tests when Sending Packets

> ⚠️ ***Notice:***
> *When testing power consumption of sending packets on ESP-Launcher, the baud rate is 115200.*

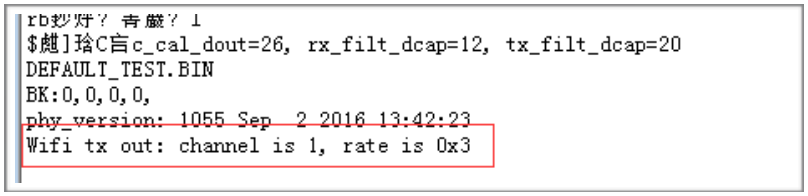
1. Packet send operation power consumption test #1:

    Power consumption test when ESP-Launcher sends packets in the 802.11b mode, CCK 11 Mbps, Pout = +17 dB.

    - Input the following command via UART to run packet send operation power consumption test #1:

```
esp_tx_50 1 0x3
```

    - After execution of the command, UART outputs logs as Figure 4-3 shows:



```
$鏂]琀C言c_cal_dout=26, rx_filt_dcap=12, tx_filt_dcap=20
DEFAULT_TEST.BIN
BK:0,0,0,0,
phy_version: 1055 Sep  2 2016 13:42:23
Wifi tx out: channel is 1, rate is 0x3
```

Figure 4-3. Printed Logs of Packet Send Operation Power Consumption Test #1

- Execute the Python script to run the power consumption test on ESP-Launcher in the mode of 802.11b, CCK 11 Mbps, Pout = +17 dB. The average power consumption is 163.96 mA, as Figure 4-15 shows:

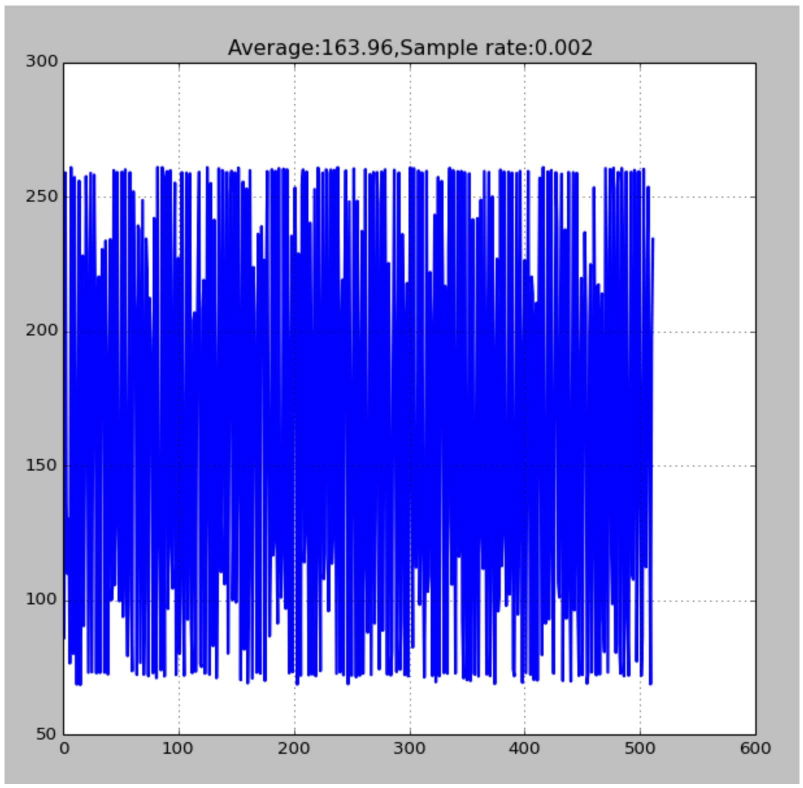
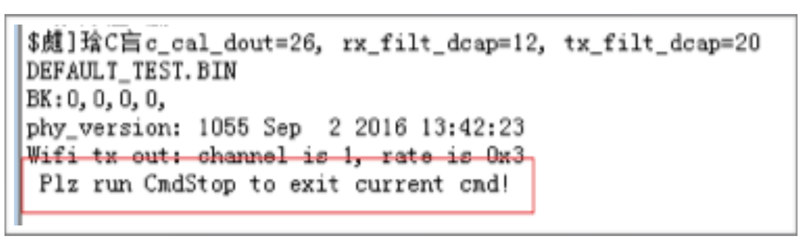


Figure 4-4. Current Measurement of Packet Send Operation Power Consumption Test #1

- Input the following command via UART to exit packet send operation power consumption test #1

```
cmdstop
```

- UART outputs the following message:



```
$鏂]琀C言c_cal_dout=26, rx_filt_dcap=12, tx_filt_dcap=20
DEFAULT_TEST.BIN
BK:0,0,0,0,
phy_version: 1055 Sep  2 2016 13:42:23
Wifi tx out: channel is 1, rate is 0x3
 Plz run CmdStop to exit current cmd!
```

Figure 4-5. Printed Logs on Exiting the Test

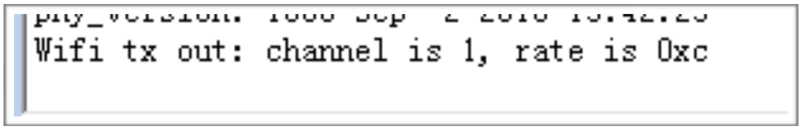
2. Packet send power operation consumption test #2:

   Power consumption test when ESP-Launcher sends packets in the 802.11g mode, OFDM 54 Mbps, Pout = +15 dB.

   - Input the following command via UART to run packet send operation power consumption test #2:

```
esp_tx_50 1 0xc
```

   - UART outputs the following message:



```
Wifi tx out: channel is 1, rate is 0xc
```

Figure 4-6. Printed Logs of Packet Send Operation Power Consumption Test #2

   - Execute the Python script to run the power consumption test on ESP-Launcher in the 802.11g mode, OFDM 54 Mbps, Pout = +15 dB. The average power consumption is 143.77 mA, as Figure 4-7 shows:

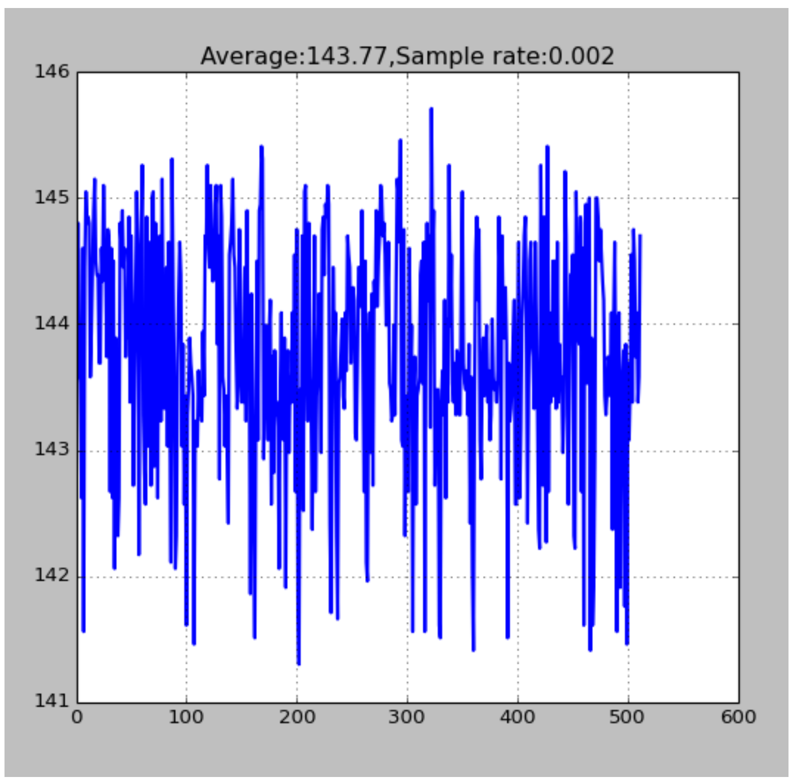


Figure 4-7. Current Measurement of Packet Send Operation Power Consumption Test #2

   - Input the following command via UART to exit packet send operation power consumption test #2:

```
cmdstop
```

3. Packet send operation power consumption test #3:

Power consumption test when ESP-Launcher sends packets in the mode of 802.11n, MCS7, Pout = +13 dB.

- Input the following command via UART to run packet send operation power consumption test #3:

```
esp_tx_50 1 0x17
```

- After execution of the command, UART outputs log as Figure 4-8 shows:

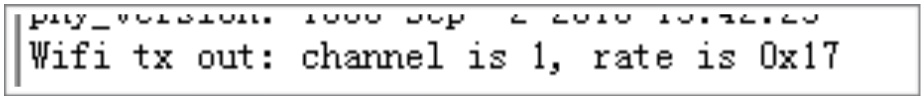


Figure 4-8. Printed Logs of Packet Send Operation Power Consumption Test #3

- Execute the Python script to run the power consumption test on ESP-Launcher in the mode of 802.11n, MCS7, Pout = +13 dB. The average power consumption is 112.66 mA, as Figure 4-9 shows:

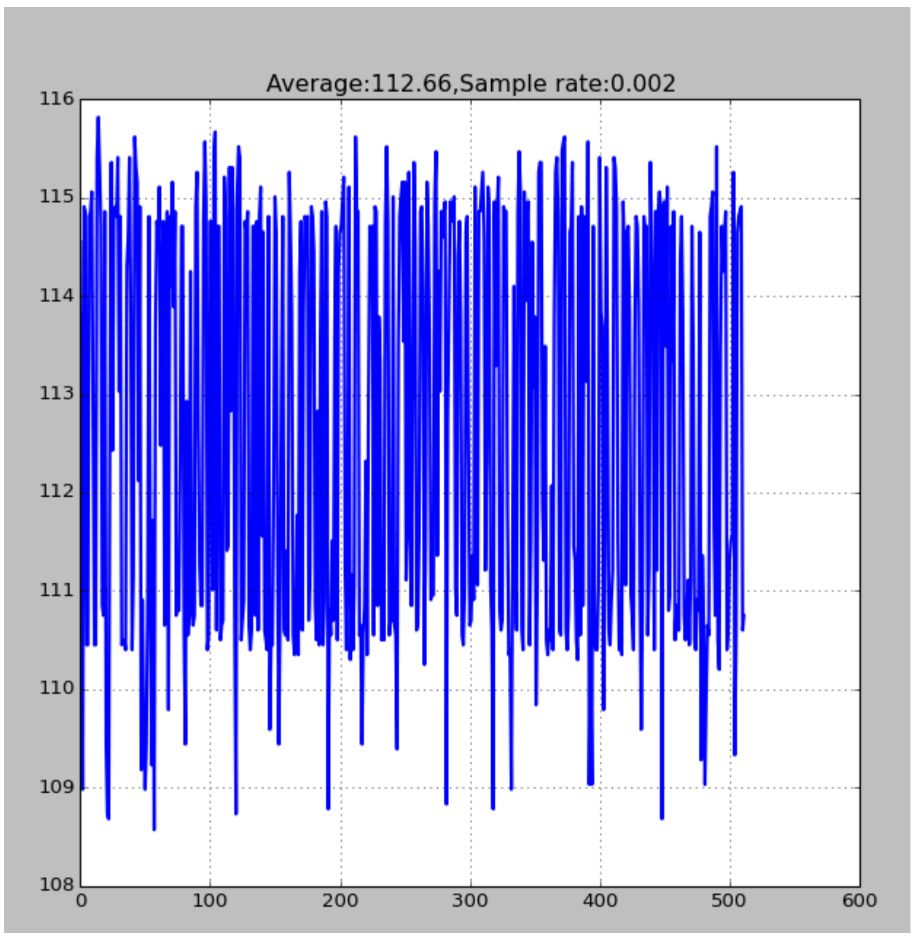


Figure 4-9. Current Measurement of Packet Send Operation Power Consumption Test #3

- Input the following command via UART to exit packet send operation power consumption test #3:

```
cmdstop
```

4.3. Power Consumption Tests when Receiving Packets

⚠ ***Notice:***

- *When testing the power consumption of ESP8266 when receiving packets, the baud rate should be 115200.*
- *When testing the power consumption of ESP8266 when receiving packets, use a WT-200 Wireless Tester as a signal source to send corresponding signals to the module.*
- *According to the setup and environment, the path and propagation losses for the test signal should be compensated for at the RF source (WT-200). In this case, the loss is 20 dB, therefore the source has been set to 20 dB higher to compensate.*

1. Packet receive operation power consumption test #1:

   Power consumption test when ESP-Launcher receives 1024 bytes long packets in an environment of IEEE 802.11g standard and -80 dB transmit power.

   - Configure the VSG parameters in the WLAN Meter. Set ***Signal Type*** to 1 Mbps and define ***Repeat Count*** as 1000. WT-200 configuration for the test environment is shown in Figure 4-10.

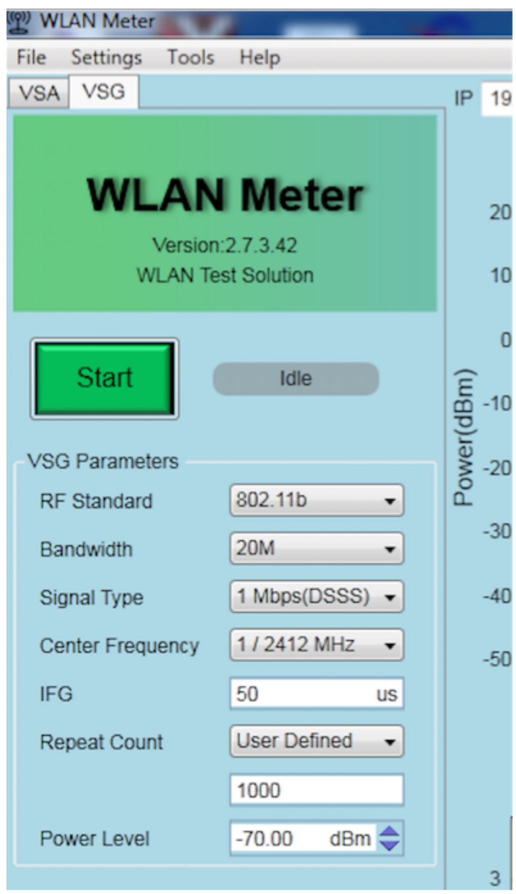


Figure 4-10. WT-200 Configuration for Test Environment #1

- Input the following command to ESP-Launcher via the UART interface to enter packet receive operation power consumption test #1:

```
esp_rx 1 0x0
```

- After execution of the command, UART will output logs as Figure 4-11 shows. ESP8266 receives100% of all the data sent in this case.

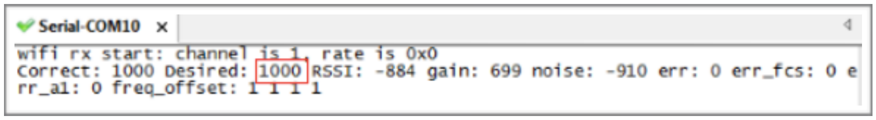


Figure 4-11. UART Output Logs of Packet Receive Operation Power Consumption Test #1

- Execute the Python script to test the power consumption when ESP-Launcher receives 1024 bytes long packets in an environment of IEEE 802.11g standard and -80 dB transmit power. The average power consumption is 71.43 mA, as Figure 4-12 shows:

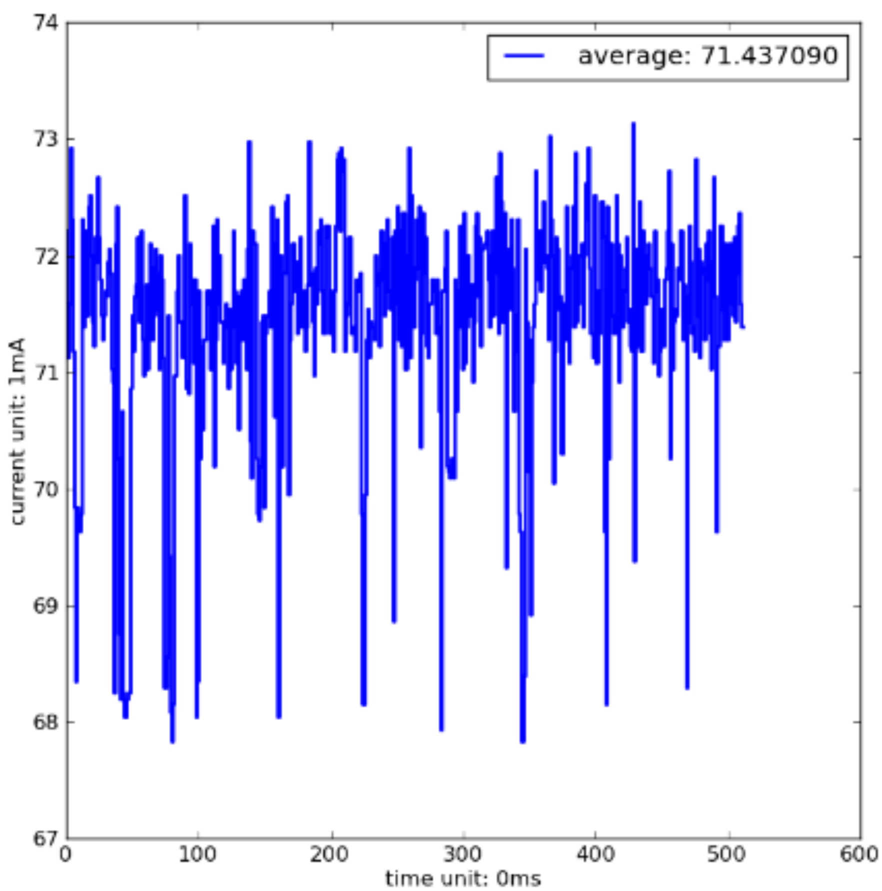


Figure 4-12. Current Measurement of Packet Receive Operation Power Consumption Test #1

2. Packet receive operation power consumption test #2:

Power consumption test when ESP-Launcher receives 1024 bytes long packets in an environment of IEEE 802.11g standard and -70 dB transmit power.

- Configure the VSG parameters in the WLAN Meter. Set ***Signal Type*** to 54 Mbps and define ***Repeat Count*** as 1000. WT-200 configuration for the test environment is shown in Figure 4-13.

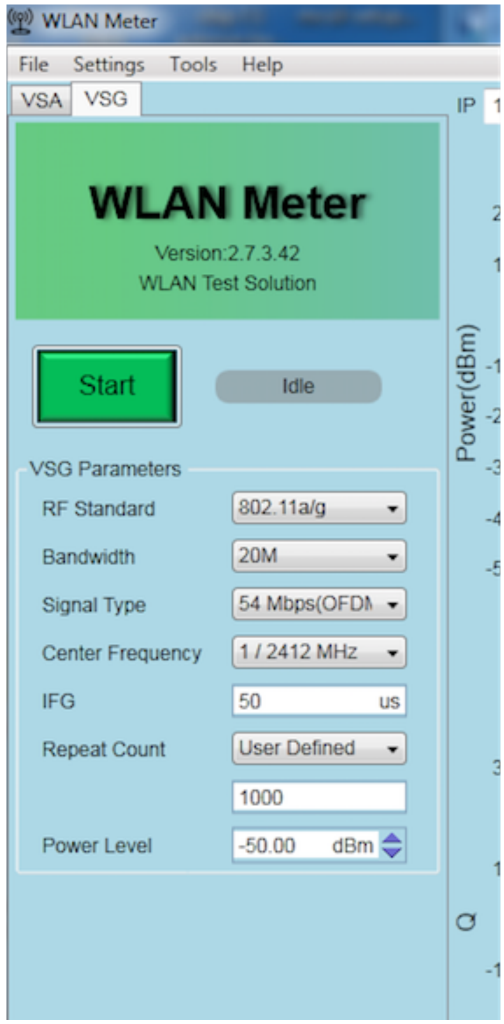


Figure 4-13. WT-200 Configuration for Test Environment #2

- Input the following command to ESP-Launcher via the UART interface to enter packet receive operation power consumption test #2:

```
esp_rx 1 0xc
```

- After execution of the command, UART will output logs as Figure 4-14 shows. ESP8266 receives 94.4% of the data sent in this case.

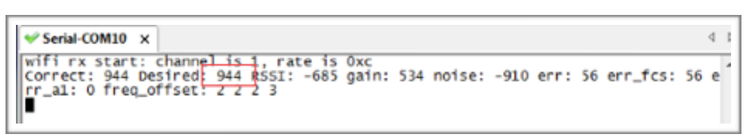


Figure 4-14. Printed Logs of Packet Receive Operation Power Consumption Test #2

- Execute the Python script to test the power consumption when ESP-Launcher receives 1024 bytes long packets in an environment of IEEE 802.11g standard and -70 dB transmit power.The average power consumption is 71.36 mA, as Figure 4-15 shows:

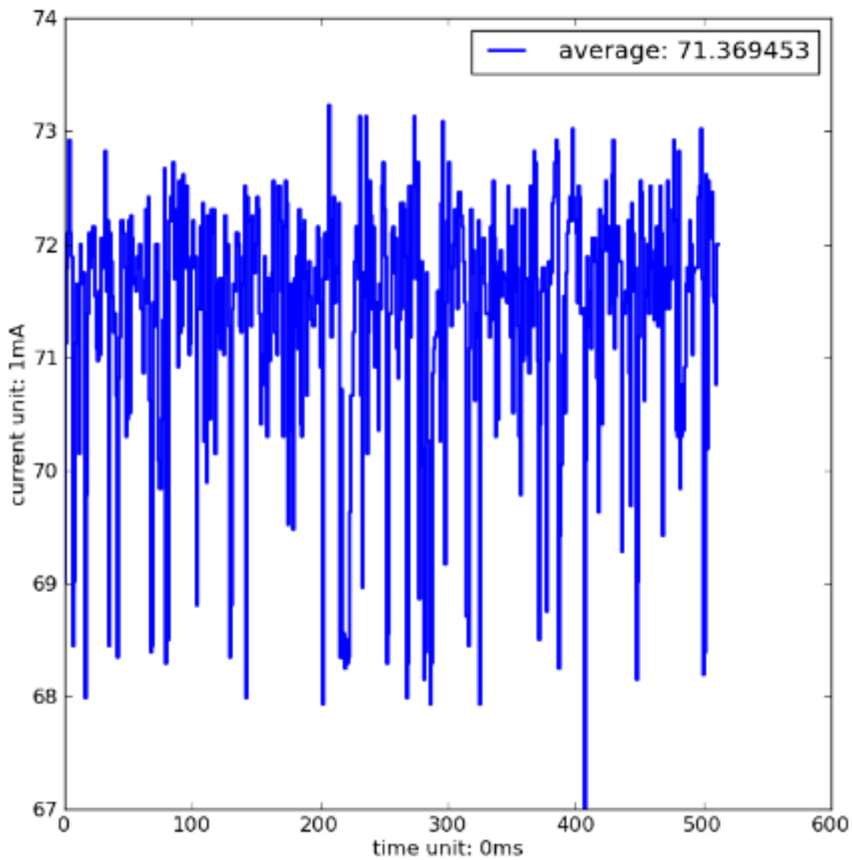


Figure 4-15. Current Measurement of Packet Receive Operation Power Consumption Test #2

3. Packet receive operation power consumption test #3:

   Power consumption test when ESP-Launcher receives 1024 bytes long packets in an environment of IEEE 802.11n standard and -65 dB transmit power.

   - Configure the VSG parameters in the WLAN Meter. Set ***Signal Type*** to MCS7 and define ***Repeat Count*** as 1000. WT-200 configuration for the test environment is shown in Figure 4-16.

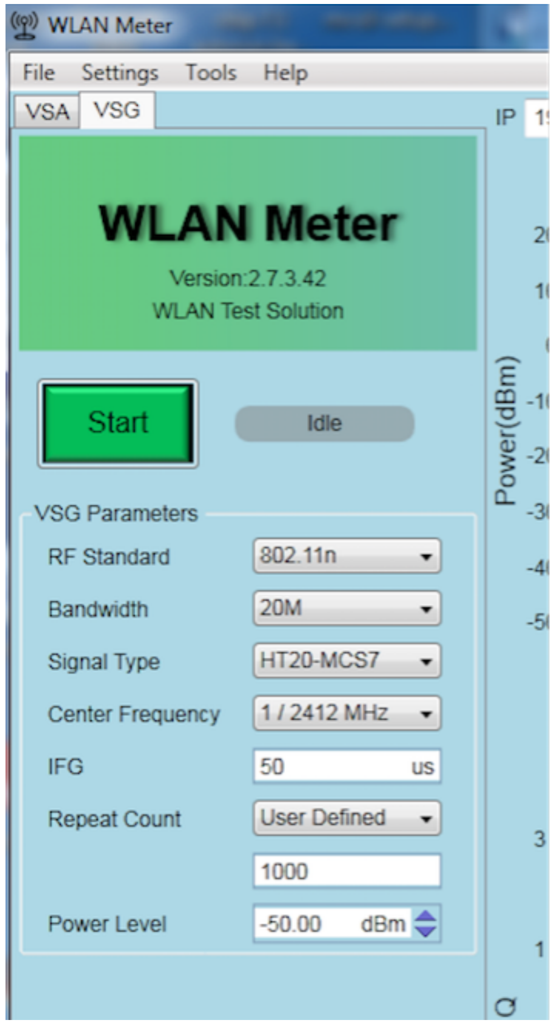


Figure 4-16. WT-200 Configuration for Test Environment #3

- Input the following command to ESP-Launcher via UART interface to enter packet receive operation power consumption test #3:

```
esp_rx 1 0x17
```

- After execution of the command, UART will output logs as Figure 4-17 shows. ESP8266 receives 93.8% of the data sent in this case.

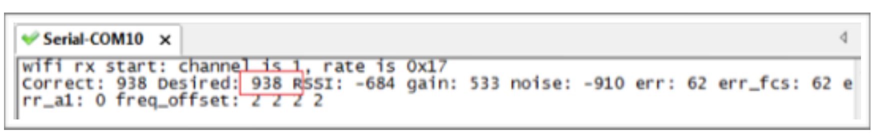


Figure 4-17. Printed Logs of Packet Receive Operation Power Consumption Test #3

- Execute the Python script to test the power consumption when ESP-Launcher receives 1024 bytes long packets in an environment of IEEE 802.11n standard and -65 dB transmit power.The average power consumption is 71.28 mA, as Figure 4-18 shows:

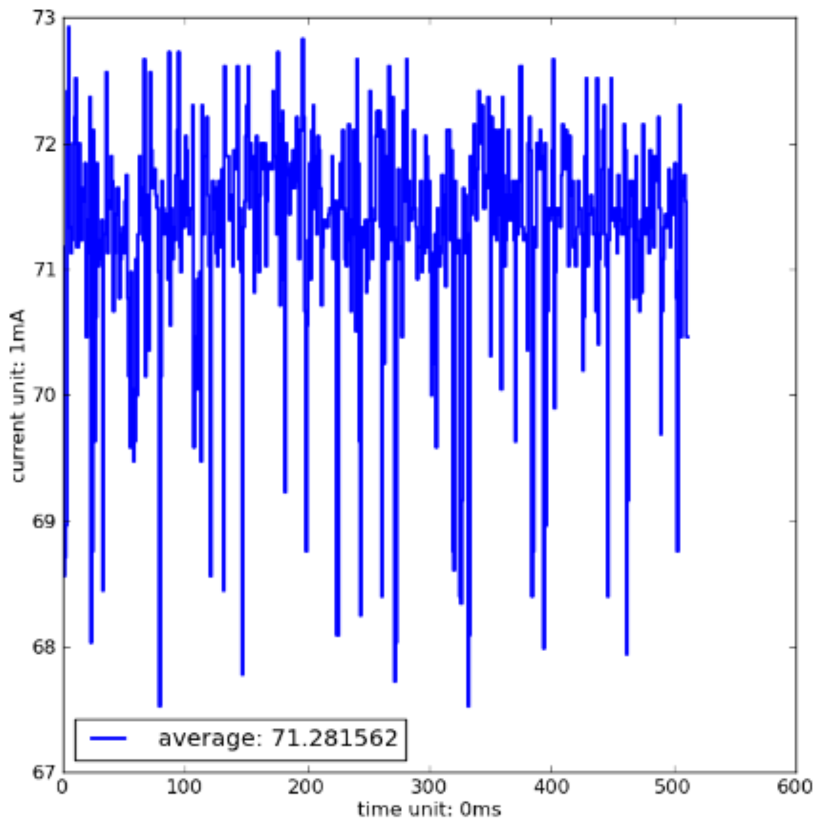


Figure 4-18. Current Measurement of Packet Receive Operation Power Consumption Test #3

5. Conclusion of the Demonstration

The conclusion of the power consumption tests conducted above in controlled environment, using various PHY modes of the ESP8266 are as follows:

1. Power consumption of ESP-Launcher in different PHY modes when sending packets:
    - Average current consumption is 163.96 mA in the 802.11b mode, CCK 11 Mbps, Pout = +17 dB.
    - Average current consumption is 143.77 mA in the 802.11g mode, OFDM 54 Mbps, Pout = +15 dB.
    - Average current consumption is 112.66 mA in the 802.11n mode, MCS7, Pout = +13 dB.
2. Power consumption of ESP-Launcher in different PHY modes when receiving packets:
    - ESP8266 receives 100% of the data with 1024 bytes per packet in an environment of IEEE 802.11b standard and -80 dB transmit power. The average power consumption is 71.43 mA.
    - ESP8266 receives 94.4% of the data with 1024 bytes per packet in an environment of IEEE 802.11g standard and -70 dB transmit power. The average power consumption is 71.36 mA.
    - ESP8266 receives 93.8% of the data with 1024 bytes per packet in an environment of IEEE 802.11n standard and -65 dB transmit power. The average power consumption is 71.28 mA.

I. Appendix - Installing UART Driver for ESP-Launcher

1. Use a Micro USB cable to connect the ESP-Launcher to the PC. The UART driver needs to be installed on the PC.

Figure I-1. Installing UART Driver on PC

Finish installing the UART driver on the PC:

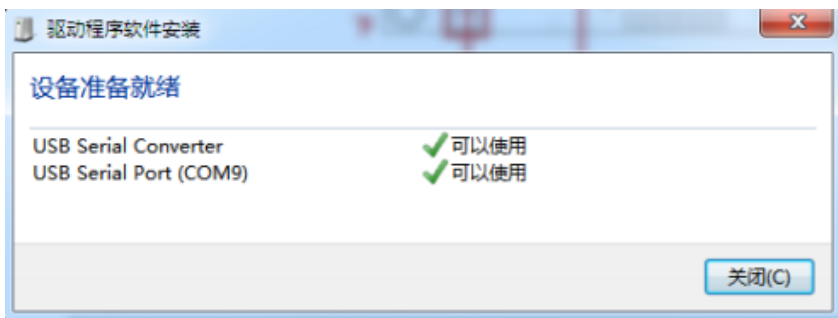


Figure I-2. UART Driver Installed

> ⚠ ***Notice:***
>
> *If the PC does not install the UART driver automatically, user can install it manually. Espressif provides the UART driver in* ***./Tools/ft232r-usb-uart.zip.***

2. Unzip ***./Tools/SecureCRT.rar***, and open the SecureCRT to check if the UART driver is installed successfully.

> 📖 ***Notes:***
>
> *The UART configuration of the ESP-Launcher should be as follows:*
>
> - *Baud rate: 74880 (by default) or 115200 (for standard AT commands)*
> - *8N1 (Data bits: 8, Parity bit: None, Stop bit: 1)*
> - *Flow control: none*
> - *The SecureCRT should be set into "New Line Mode", because AT commands are terminated with a new-line (CR-LF).*

- If it is the first time you are using the SecureCRT, create a new serial port connection. Users can set the configuration by referring to the screenshot below. Set ***Protocol*** to serial. Choose no flow control, and set the actual port number by finding out the assigned port from the "Device Manager" of PC.

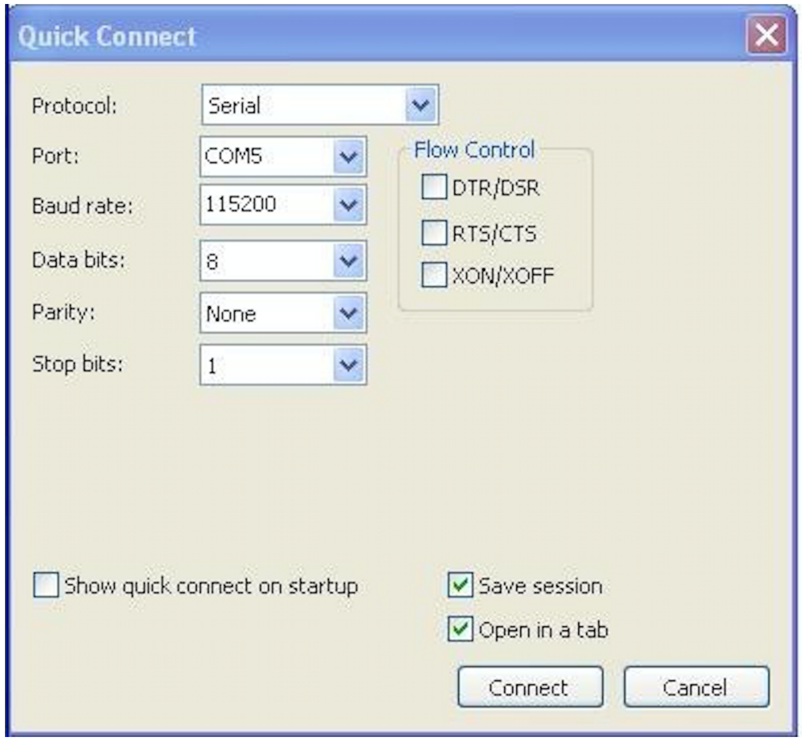


Figure I-3. Creating a New Serial Port Connection

- If the serial port connection was created before, users can change its configuration by right-clicking on it, and selecting "properties".

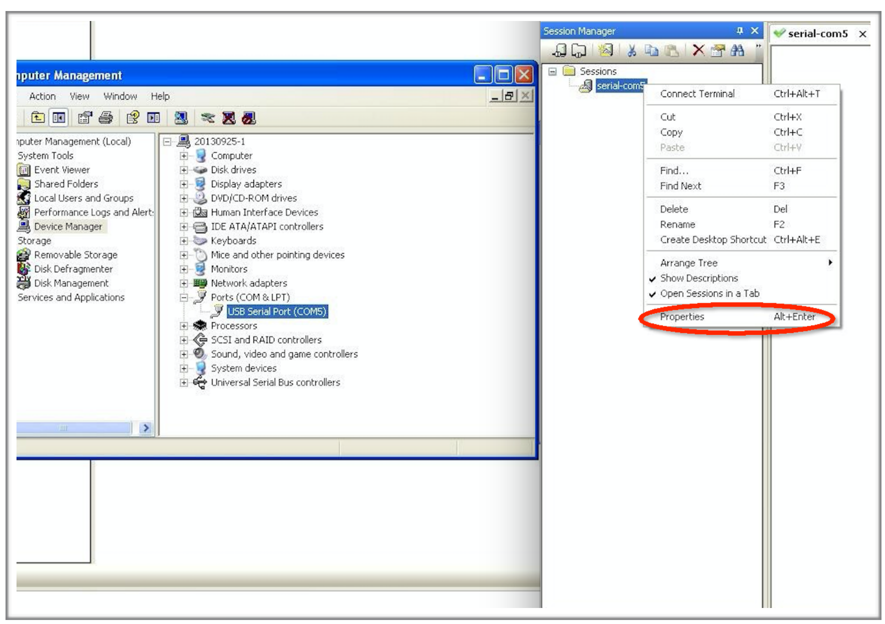


Figure I-4. Checking the "Properties" of the Serial Port

- Select the “Serial” page in the “Properties” window to check the configuration of the serial connection.

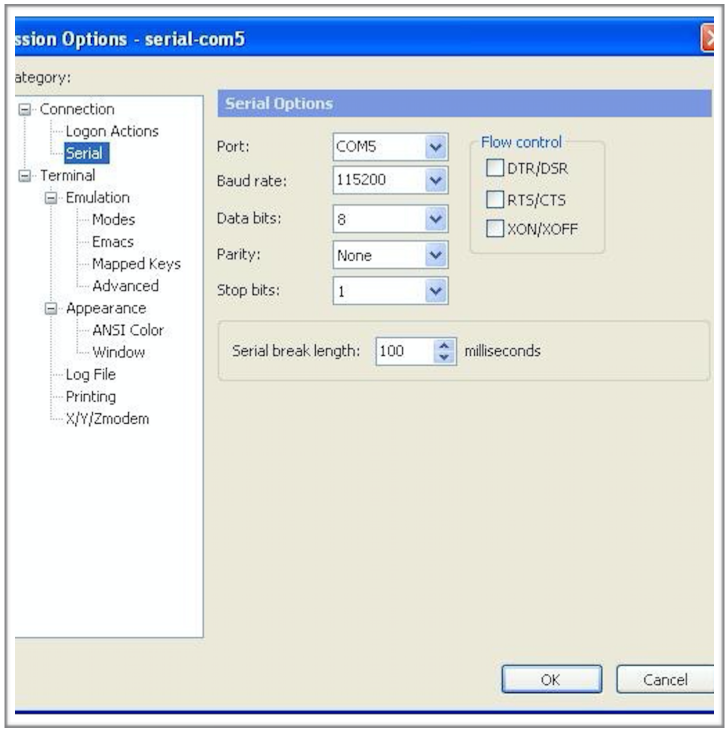


Figure I-5. Checking the “Serial” in “Properties” Window

- If AT commands are used, the baud rate should be 115200, and the “New Line Mode” should be set as the AT commands terminate with a newline (CR-LF).

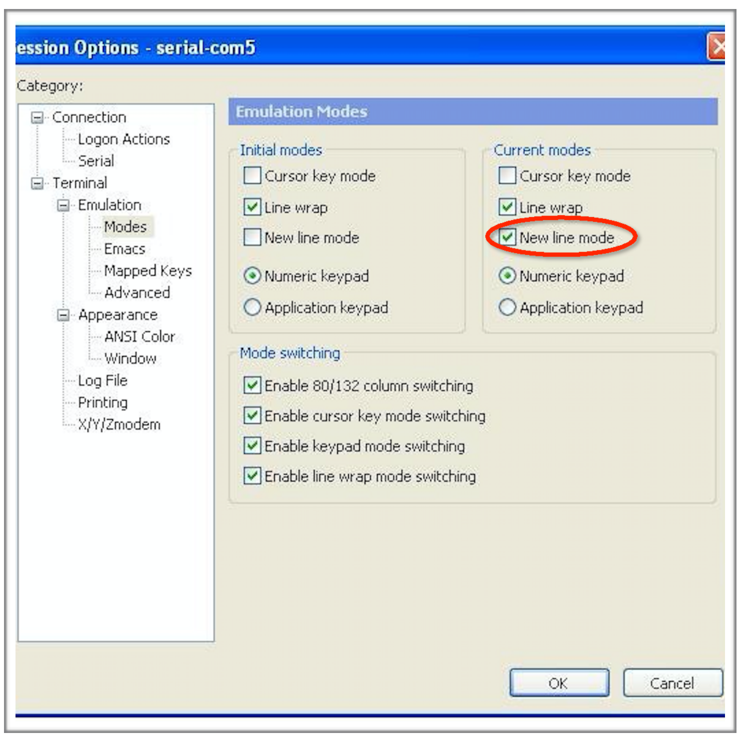


Figure I-6. Setting the “New Line Mode” for AT Commands

> **Note:**
>
> *The default baud rate of ESP8266 is 74880. But to use AT commands, the baud rate should be 115200.*

3. Power up the ESP-Launcher by toggling the switch "1" to the upper side. Check if the startup logs (baud rate 74880) are displayed on the SecureCRT.

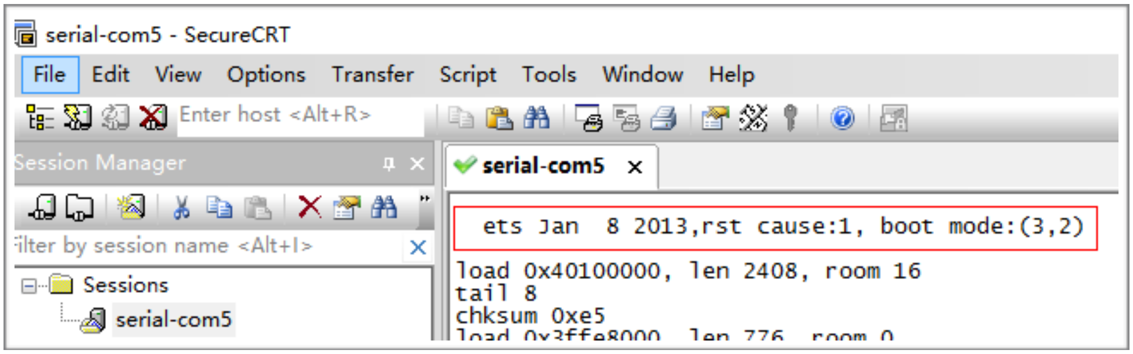
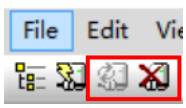


Figure I-7. Startup Logs

4. User can connect / disconnect the serial port by clicking these two buttons  at the upper left corner of the SecureCRT.

- The left button is to create a serial port connection.
- The right button is to abort the connection.

II. Appendix - Flash Download Tool Guide

1. Double click the ***./Tools/FLASH_DOWNLOAD_TOOLS_v2.4_150924/ESPFlashDownloadTool_v2.3+.exe*** to run the Flash Download Tool.
2. Fill the ***Download Path Config*** area with the location of the firmware to be downloaded, and download it to ***ADDR*** 0x0000. ***SPI SPEED*** is 40 MHz, ***SPI MODE*** is QIO, ***FLASH SIZE*** is 8 Mbit. And select the corresponding ***COM*** port.

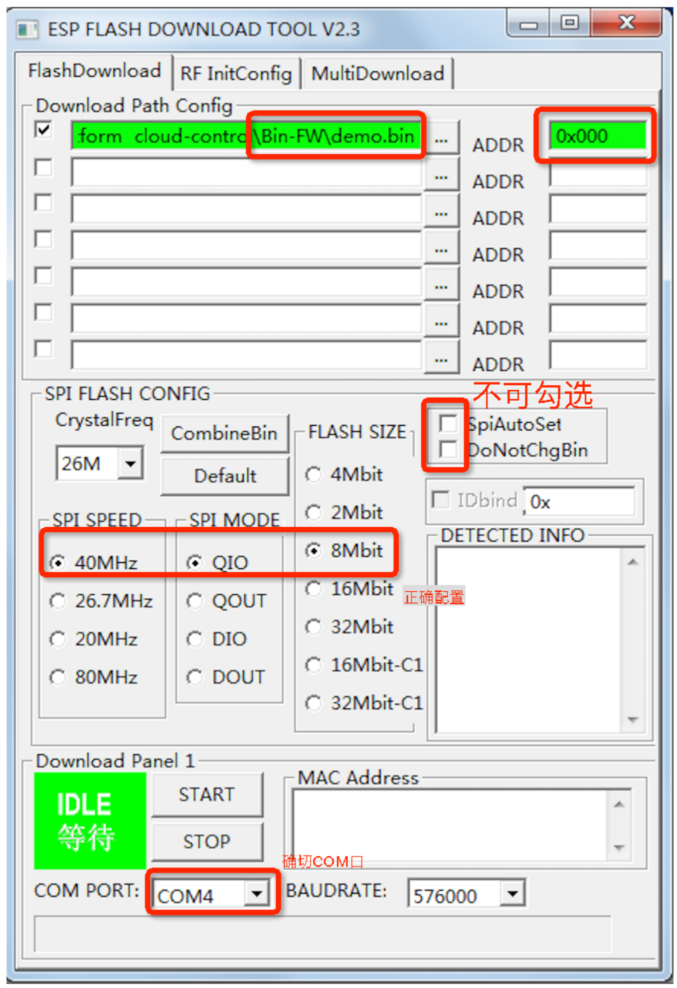


Figure II-1. Flash Download Tool

> 📖 ***Note:***
>
> *The selected firmware in the **Download Path Config** area will turn into green and be downloaded into the ESP-Launcher. If there is any other firmware which is not selected, it will be ignored.*

3. Click the "START" button on the ESP8266 Flash Download Tool to enter the "SYNC" state. Wait for the ESP-Launcher to power up.

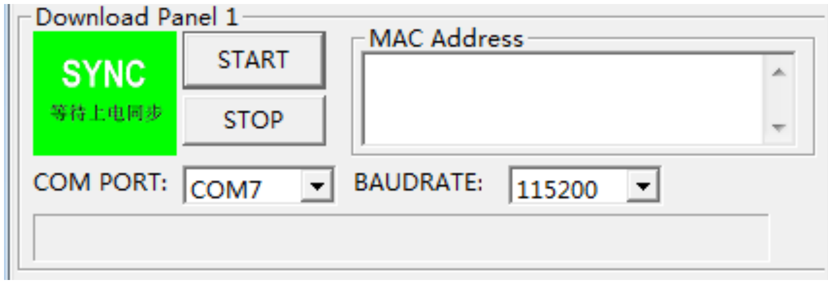


Figure II-2. "SYNC" State of Flash Download Tool

4. Power up the ESP-Launcher, and start downloading firmware.
    - Toggle the switch "2" to the lower side to set ESP-Launcher to download mode. And toggle the switch "1" to the upper side to power up the ESP-Launcher. More details about the ESP-Launcher are in ***Section 3.4***.

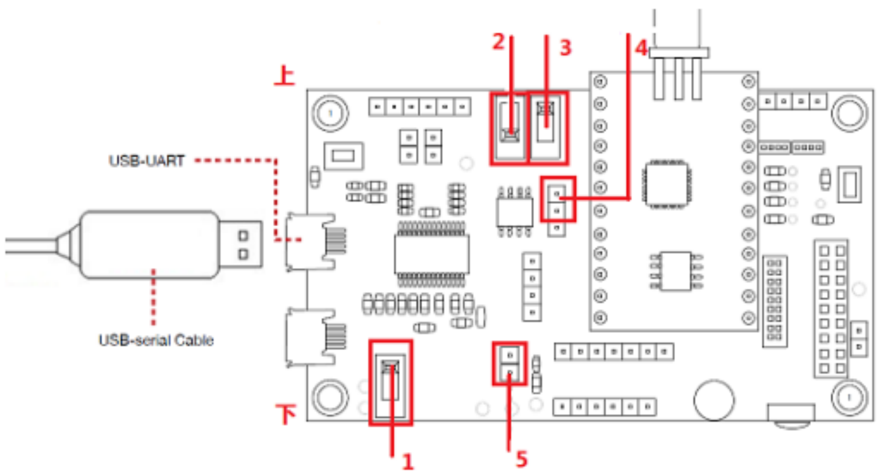


Figure II-3. ESP-Launcher

- The Flash Download Tool will start to download firmware into the ESP-Launcher. The ***DETECTED INFO*** area on the ESP8266 Flash Download Tool will display information about the flash chip on the ESP-Launcher.

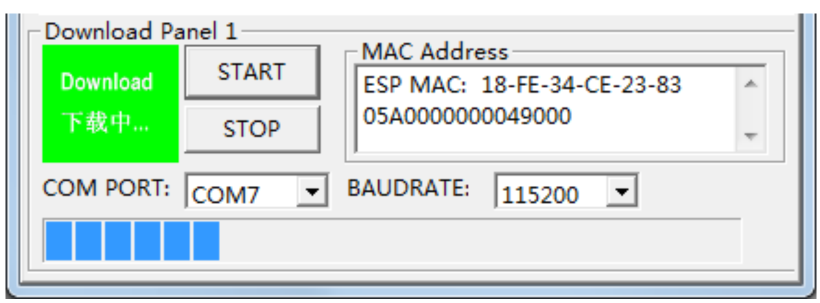


Figure II-4. Starting Downloading Firmware

- Wait for the download to complete. "FINISH" state will be shown on the Flash Download Tool.

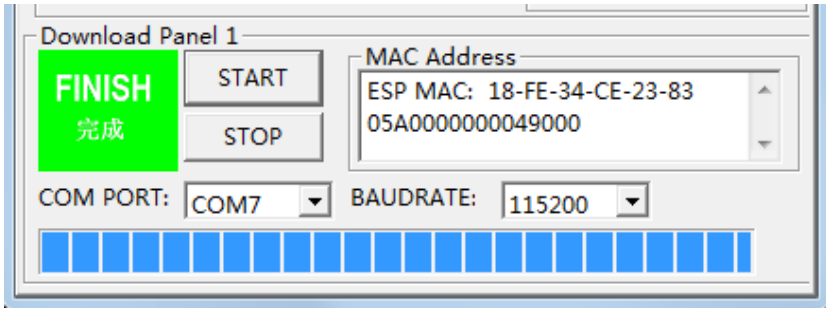


Figure II-5. Downloading Firmware Finished

After the download is finished, toggle the switch “1” to the lower position to power off the ESP-Launcher.

Set the ESP8266 to be in program execution mode by toggling the switch “2” to the upper position. Then toggle the switch “1” to the upper position again to power on the ESP-Launcher.

III. Appendix - Python Script Guide

1. The demonstration uses Agilent 34401A multimeter as the ammeter, which connects to the PC via the GPIB interface.
2. PyVISA module need to be installed in order to run the Python script.
3. After installing all the software needed, connect the USB of the ammeter to the PC.
4. Download the tested firmware to the ESP8266 module. Execute the firmware on the module and set the baud rate to 115200.
5. Now begin the current measurement. Adjust the ammeter to show the value of the current. Run the Python script to measure the current during the tests.

Espressif IOT Team

www.espressif.com