ESP8266 Low Power Test Demonstration

Version 1.0

About This Guide

The document is structured as follows.

Chapter	Title	Content
Chapter 1	Overview	Introduction to the low power test demonstration.
Chapter 2	Organization of the Demonstration	Introduction to the organization of the low power test demonstration.
Chapter 3	Environment Setup	Setup the hardware and software environment for running the low power test demonstration.
Chapter 4	Download Procedure	User guide of the low power test demonstration.
Chapter 5	Conclusion of the Demonstration	Conclusion of the low power test demonstration.
Appendix I	Installing UART Driver for 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 User Guide	Introduction to the Espressif Flash Download tool, and how to use it to download the firmware into the ESP-Laucher.
Appendix III	User Guide of Python Script	Introduction to the usage of the python script.

Release Notes

Date	Version	Release notes
2016.09	V1.0	Initial release.

Table of Contents

1. Overview

The ESP8266 implements many low power modes to facilitate power saving and optimization in portable, battery powered applications as well as high efficiency consumer equipment such as smart light. This guide is intended to help implement and test the available low power modes and observe the resulting current consumption of the ESP8266 easily and quickly. It demonstrates the Modem-sleep mode, Light-sleep mode, and Deep-sleep mode.

The low power test has been demonstrated and run on the ESP-Launcher, ESP8266 board designed by Espressif. However, with appropriate build settings based on your application flash configuration, the test can be made to run on any ESP8266 based system.

> 📖 ***Note:***
>
> *More documentation on ESP8266, please visit:*
>
> *http://www.espressif.com/en/support/download/documents?keys=&field_type_tid%5B%5D=14.*

2. Organization of the Demonstration

To make it easy for the developer to evaluate the power consumption of hardware built around the ESP8266, we take up a UART command based approach for activating various sleep modes. This makes it easy to test the overall circuit power consumption without this test being dependent on your compiler/environment setup. The guide lists the pre-programmed UART commands used to activate the low power modes and measure their current consumption profiles.

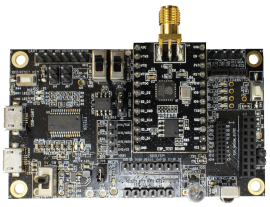
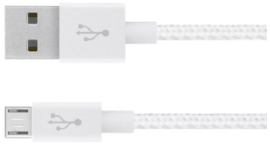
1. Modem-sleep mode:
   - Send command `workmode:1` through UART to put ESP8266 into Modem-sleep mode.
   - Send command `wakemode:1` through UART to wake ESP8266 up from Modem-sleep mode.
2. Light-sleep mode:
   - Send command `workmode:2` through UART to put ESP8266 into Light-sleep mode.
   - Pull ***GPIO12*** low to wake ESP8266 up from Light-sleep mode.
3. Deep-sleep mode:
   - Send command `workmode:3` through UART to put ESP8266 into Deep-sleep mode
   - A low pulse on ***EST-RSTB*** pin can wake ESP8266 up from Deep-sleep mode.

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 ESP8266 demo board 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 current consumption.
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 ESP8266 demo board
SecureCRT.rar	***./Tools/***	PC UART terminal emulator tool (support 74880 baud rate)
ft232r-usb-uart.zip	***./Tools/***	USB - UART converter driver
Python Script	***./Tools/***	Set the ammeter to measure the current value (python 2.7)
NI4882_1500f0.exe	***./Tools/***	Driver of the ammeter (GPIB interface)
Flash Download Tool	***./Tools/***	To download the firmware into the ESP8266 demo board

3.2. Hardware Connection

- ESP8266 demo board connects to the PC via 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 GPIB interface.
- The ammeter should be connected into the circuit by serial lines, between the ESP8266 demo board 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.

3.3. Software Installation

- Flash Download Tool, to download firmware into the ESP8266 demo board.

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

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

3.4. The ESP-Launcher

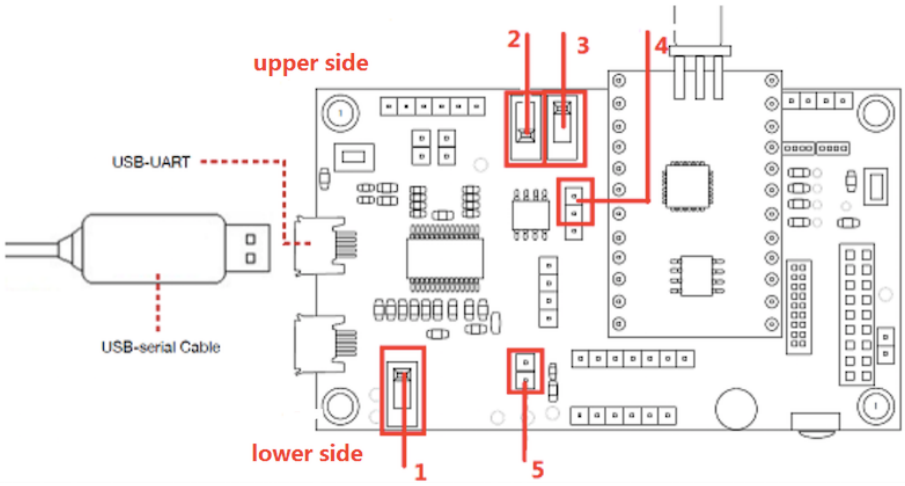


Figure 3-1. the ESP-Launcher

The switch “1”: toggle to the lower side

- Lower side: power-off
- Upper side: power-on

The switch “2”: toggle to the lower side

- Lower side: downloading firmware 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 side.

- Lower side: chip disable
- Upper side: chip enable

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

The pin “5”: put a jumper cap on it.

4. Download Procedure

4.1. Downloading Firmware

1. Download firmware into 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***.

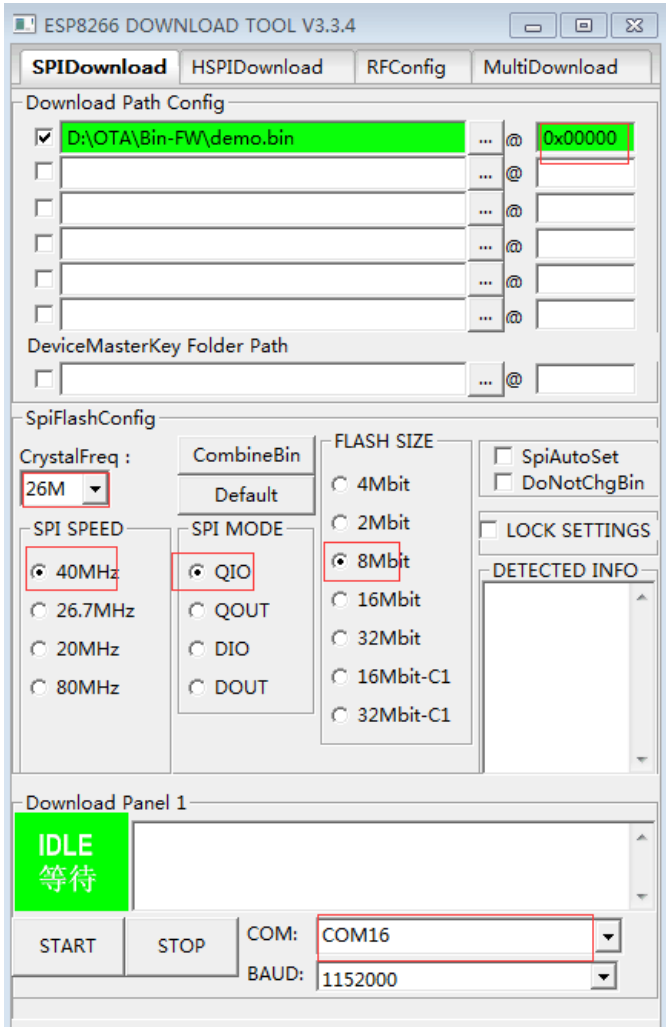


Figure 4-1. ESP8266 DOWNLOAD TOOL Interface

> 📖 ***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 was successful.
   - Connect the ESP8266 demo board to the PC using the Micro USB cable.
   - Open the UART serial monitor tool on your PC, and configure the serial terminal to ***74880-8-n-1***.
   - Toggle the switch "2" on the ESP8266 demo board to the upper side to configure it to be in program execution mode on power up.
   - Next, toggle the switch "1" to the upper side to power up the ESP8266 demo board.
   - If the UART log output is as shown below, it means that the firmware is running successfully.

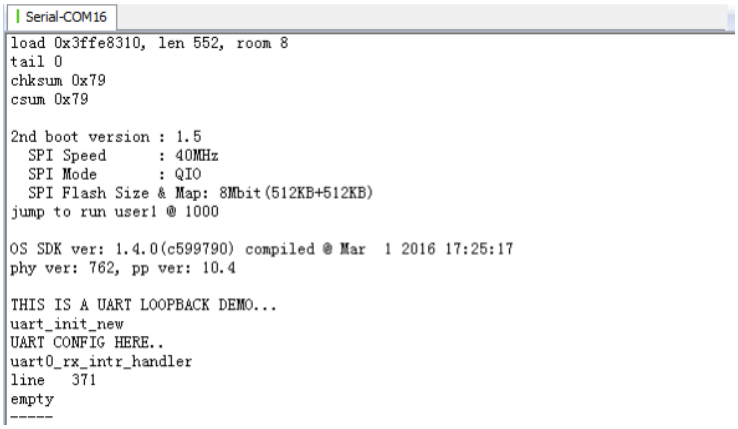


Figure 4-2. UART Output Logs

4.2. Modem-sleep Test

1. Input command `workmode:1` through UART to set ESP8266 into Modem-sleep mode. After ESP8266 enters Modem-sleep mode successfully, the following message will be printed as shown below.

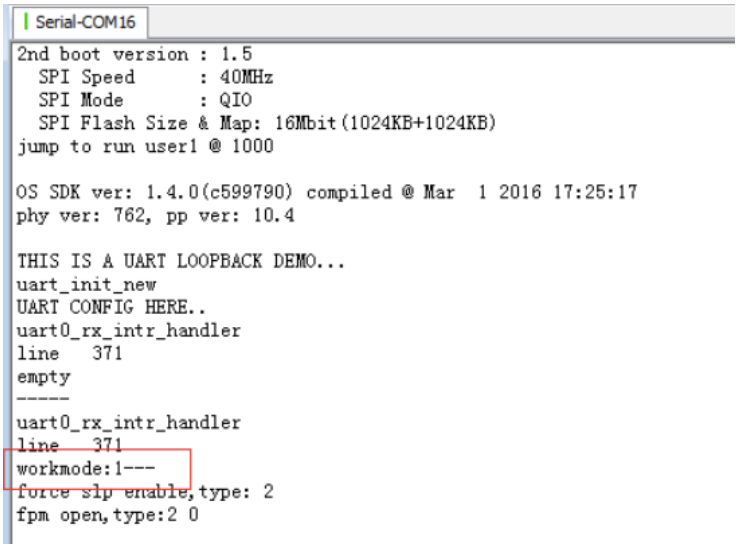


Figure 4-3. Modem-sleep Mode Test Logs

2. Run the python script to measure the current of the ESP8266 in Modem-sleep mode. The test result is shown as below, the average current is 17.88 mA.

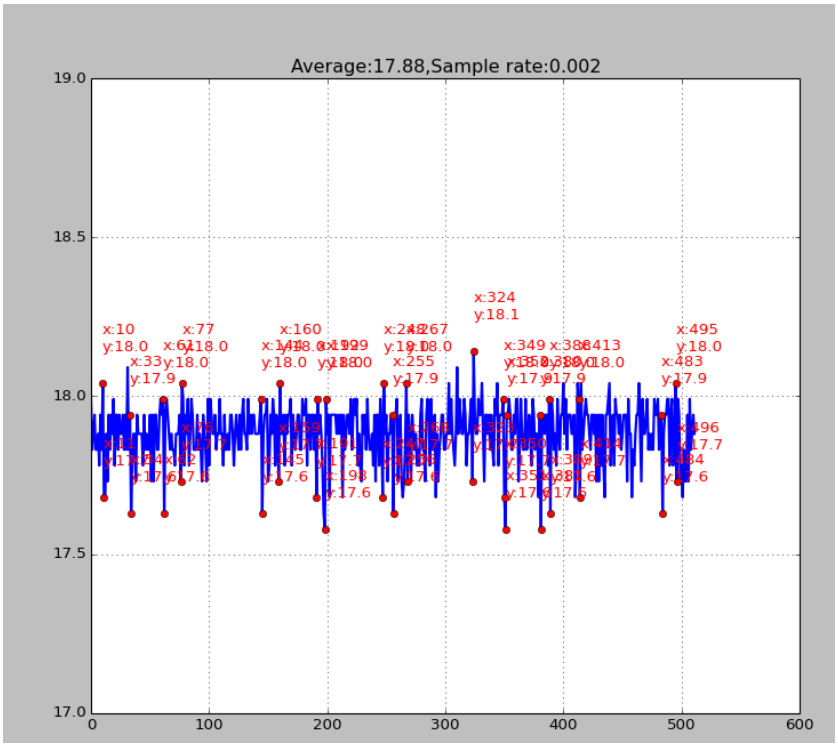


Figure 4-4. Average Current of the ESP8266 in Modem-sleep Mode

3. Input command `wakemode : 1` through UART to wake ESP8266 up from Modem-sleep mode. After ESP8266 wakes up from Modem-sleep mode successfully, a message will be printed as shown below.

```
Serial-COM16
jump to run user1 @ 1000

OS SDK ver: 1.4.0(c599790) compiled @ Mar  1 2016 17:25:17
phy ver: 762, pp ver: 10.4

THIS IS A UART LOOPBACK DEMO...
uart_init_new
UART CONFIG HERE..
uart0_rx_intr_handler
line   371
empty
-----
uart0_rx_intr_handler
line   371
workmode:1---
force slp enable,type: 2
fpm open,type:2 0
uart0_rx_intr_handler
line   371
wake:mode1---
fpm close 3
```

Figure 4-5. Wake ESP8266 up from Modem-sleep Mode

4. Run the python script to measure the current of the ESP8266 after it wakes up from Modem-sleep mode. The test result is shown below, the average current is 71.04 mA.

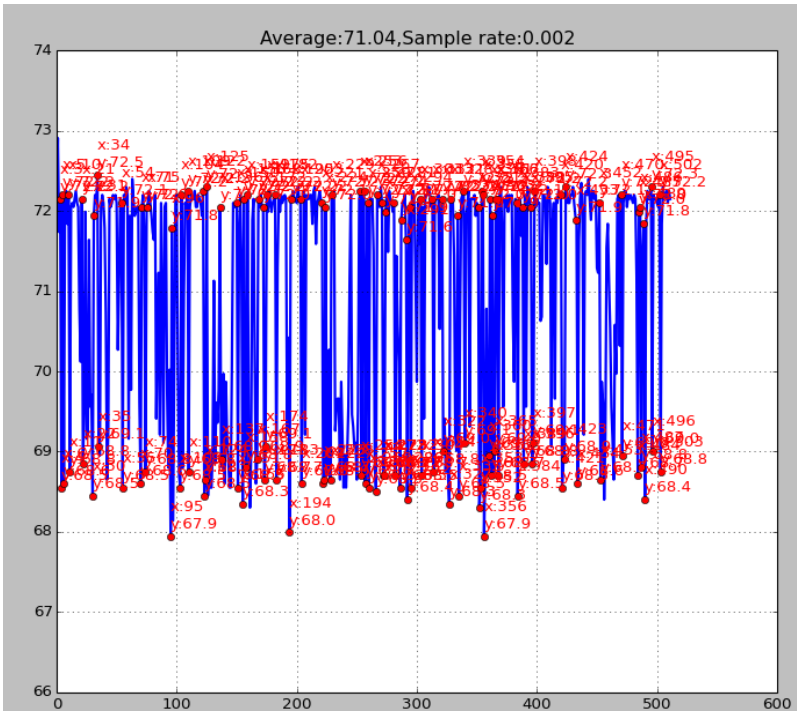


Figure 4-6. Average Current Consumption of the ESP8266 after Waking up from Modem-sleep Mode

4.3. Light-sleep Test

1. Input command `workmode:2` through UART to set ESP8266 into Light-sleep mode. After the ESP8266 enters Light-sleep mode successfully, a message will be printed as shown below.

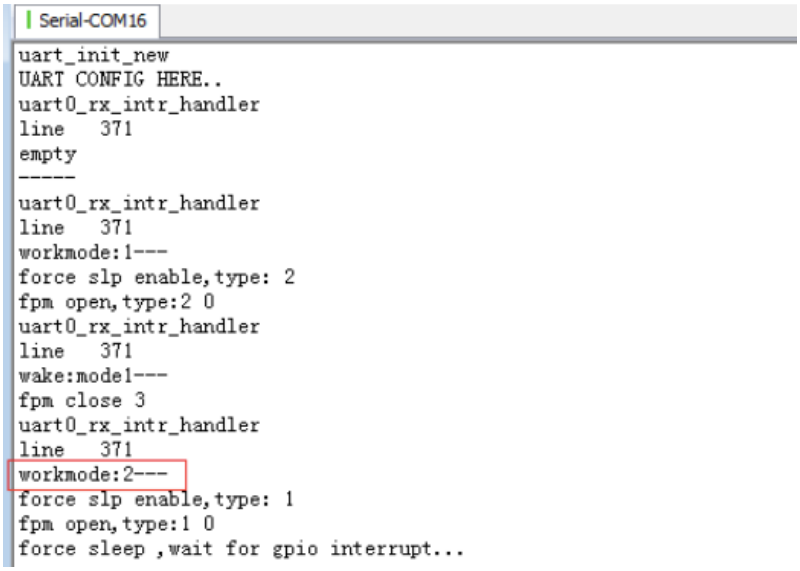


Figure 4-7. Light-sleep Mode Test Logs

2. Run the python script to measure the current of the ESP8266 in Light-sleep mode. The test result is shown below, the average current is 1.11 mA.

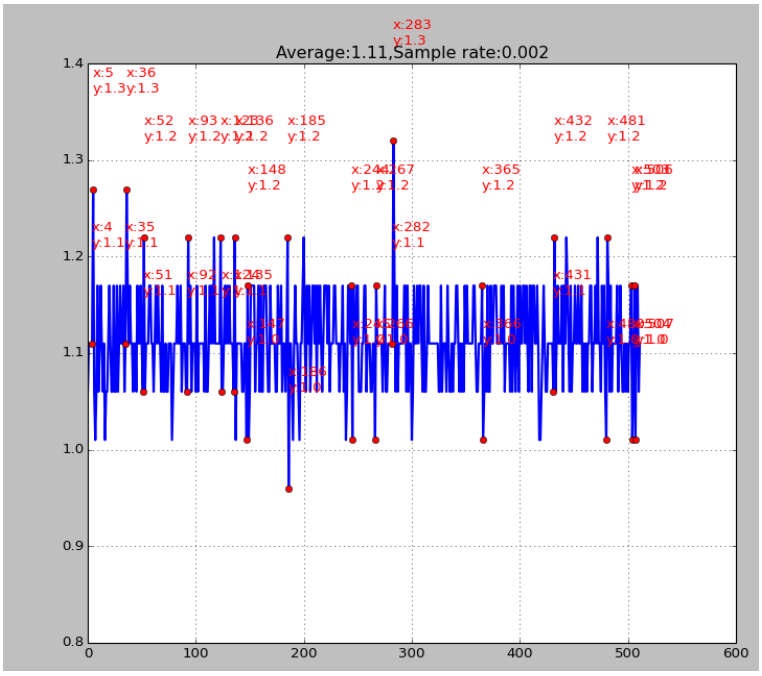


Figure 4-8. Average Current of the ESP8266 in Light-sleep Mode

3. Pull ***GPIO12*** low to wake ESP8266 up from Light-sleep mode. Measure the current after ESP8266 wakes up from Light-sleep mode successfully, the average current is 70.88 mA, as it should be in active mode.

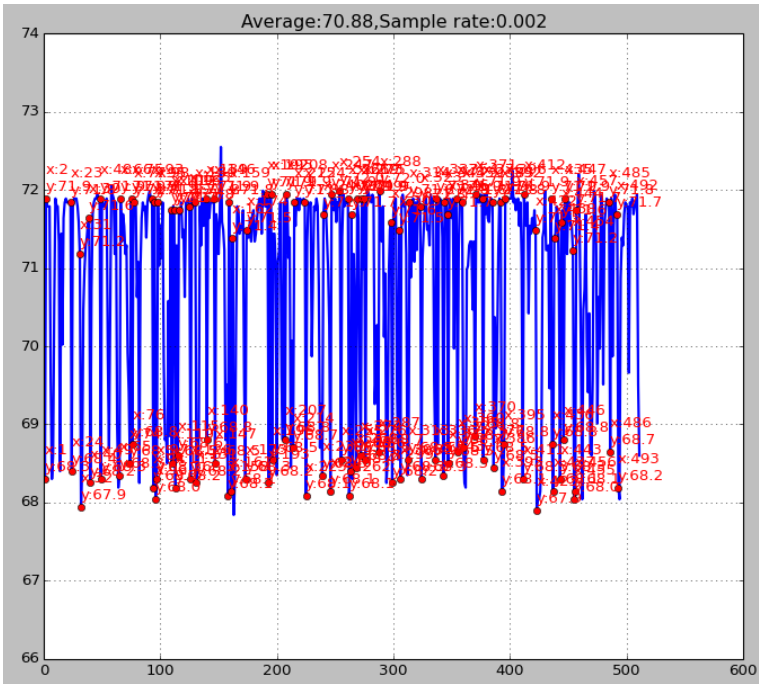


Figure 4-9. Average Current Consumption of the ESP8266 after Waking up from Light-sleep Mode

4.4. Deep-sleep Test

1. Input command `workmode:3` through UART to set ESP8266 into Deep-sleep mode. Measure the current after ESP8266 enters Deep-sleep mode. Note that the current consumption is too low to be measured by conventional current measurement equipment. You may need a benchtop multimeter for measuring the current. The average current observed is 0.0182 mA.

Figure 4-10. Average Current of ESP8266 in Deep-sleep Mode

2. Pull ***EXT_RSTB*** low to wake ESP8266 up from Deep-sleep mode. Measure the current after ESP8266 wakes up from Deep-sleep mode successfully, the average current is 71.03 mA, as it should be in active mode.

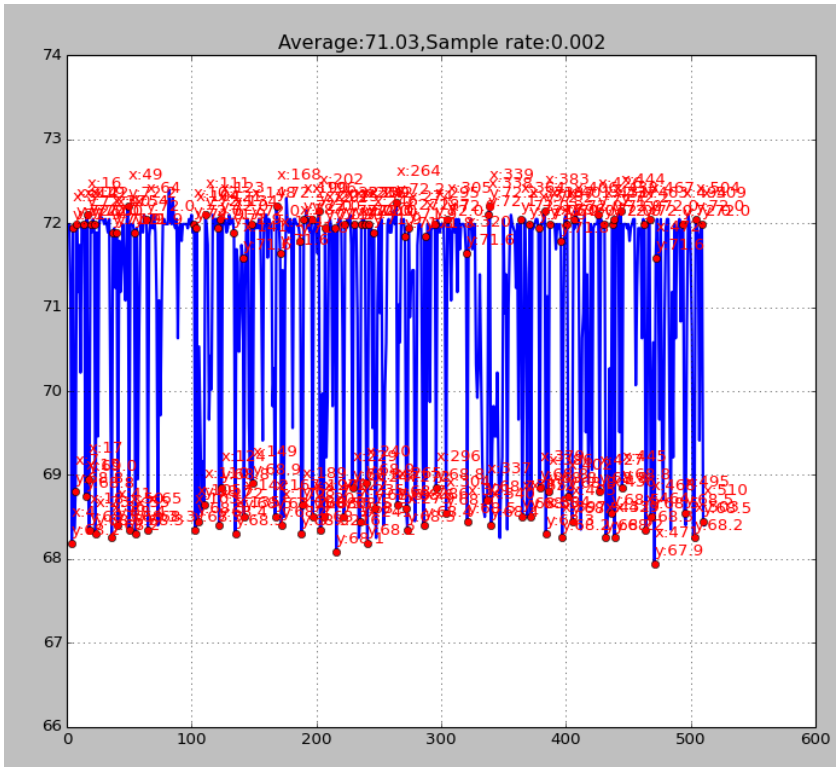


Figure 4-11. Average Current Consumption of the ESP8266 after Waking up from Deep-sleep Mode

4.5. Power-Off Test

Power off the ESP8266 demo board to measure the error value of the ammeter. The test result is shown below. Compensate for this zero error accordingly.

Figure 4-12. Average Current when ESP8266 is Powered off (to be treated as zero-error)

5. Conclusion of the Demonstration

Conclusion of the test results:

- When the ESP8266 works normally, the average current is about 71 mA.
- When the ESP8266 is in Modem-sleep mode, the average current is about 17.88 mA.
- When the ESP8266 is in Light-sleep mode, the average current is about 1.11 mA.
- When the ESP8266 is in Deep-sleep mode, the average current is about 0.018 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:

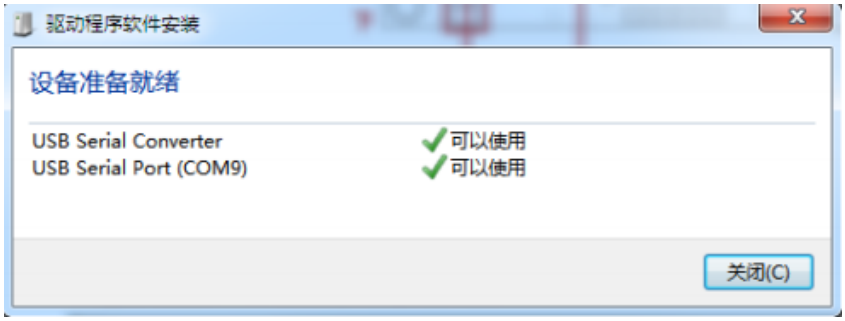


Figure I-2. UART Driver Installed

> ⚠ ***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 AT commands)*
> - *8N1 (Data bits: 8, Parity bit: None, Stop bit: 1)*
> - *Flow control: disabled*
> - *If AT commands are used, the SecureCRT should be set into "New Line Mode", because that AT commands are ended with a new-line (CR-LF).*

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

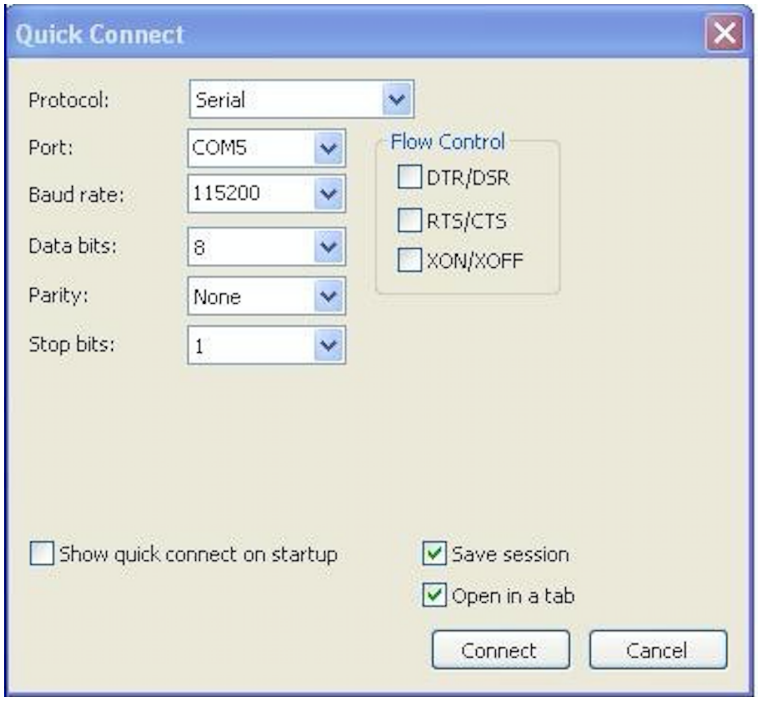


Figure I-3. Creating a New Serial Port Connection

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

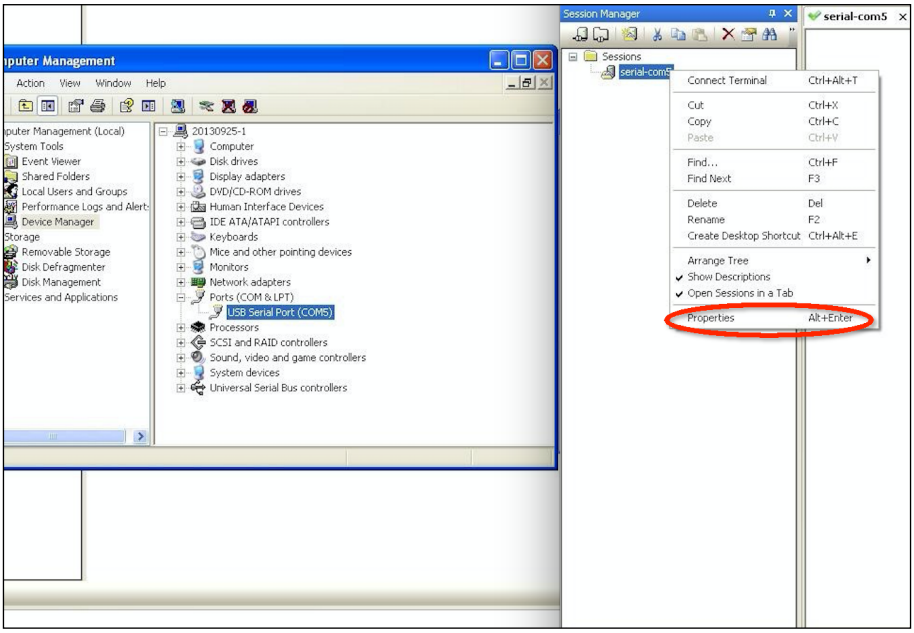


Figure I-4. Checking the “Properties” of the Serial Port

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

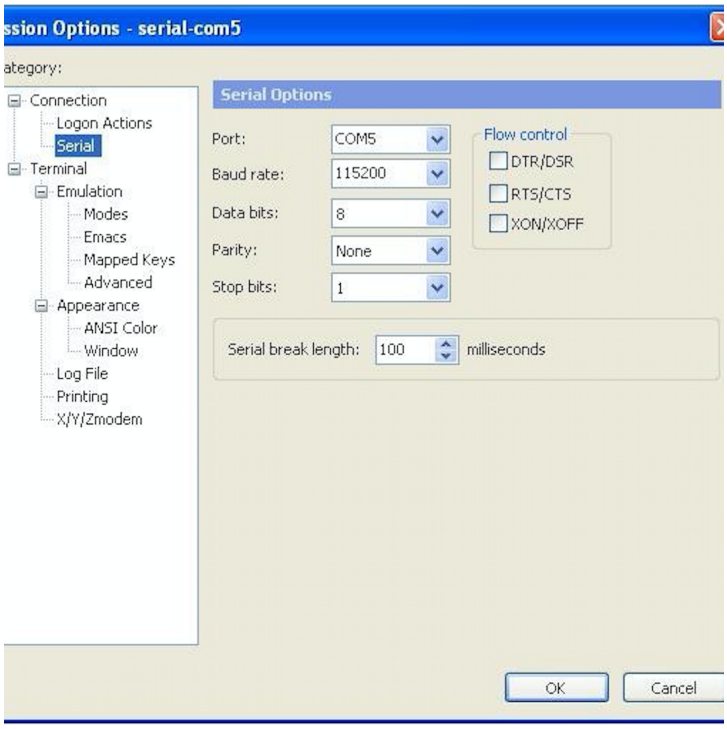


Figure I-5. Checking the “Serial” in “Properties” Window

- If AT commands are used, the baud rate should be 115200, and the “New Line Mode” should be set because the AT commands end with a new line (CR-LF).

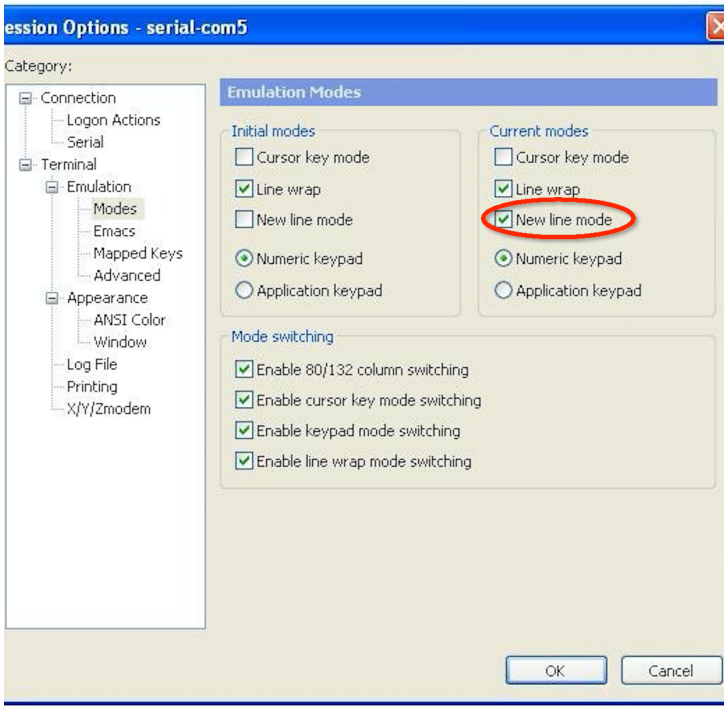


Figure I-6. Setting the “New Line Mode” for AT Commands

> **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.

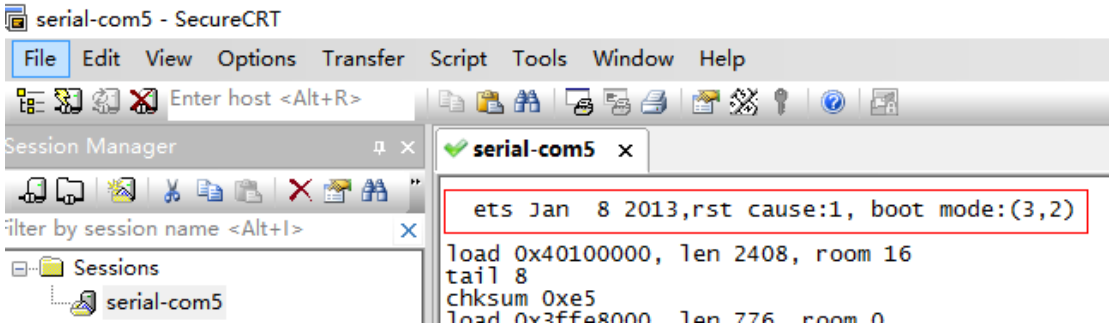


Figure I-7. Startup Logs

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 User 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 which is 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.

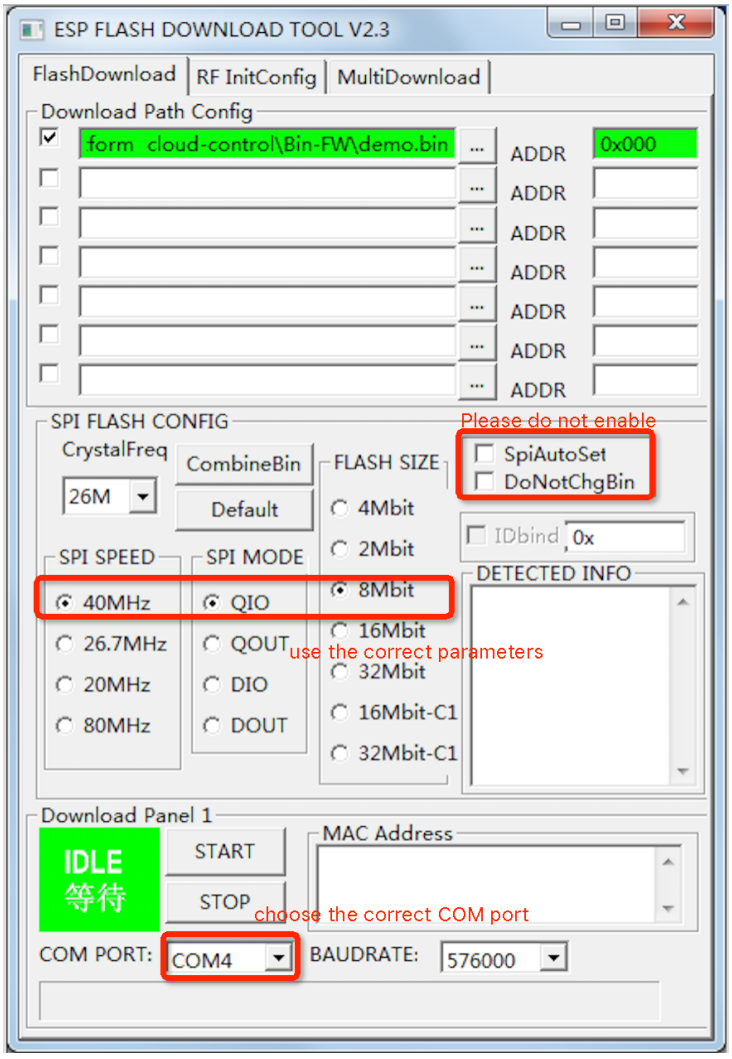


Figure II-1. ESP FLASH DOWNLOAD TOOL

> 📖 ***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.

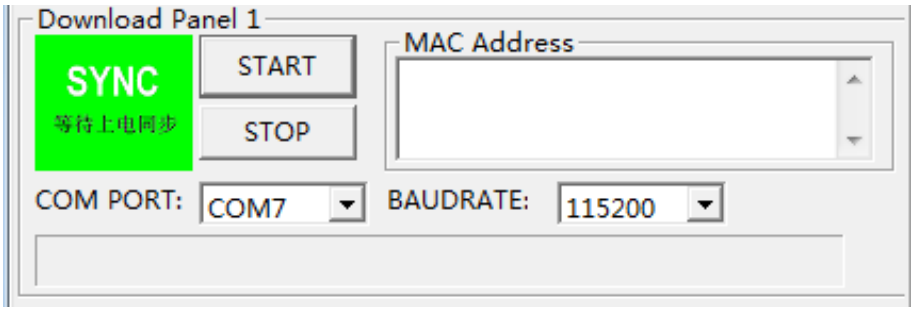


Figure II-2. “SYNC” State of Flash Download Tool

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***.

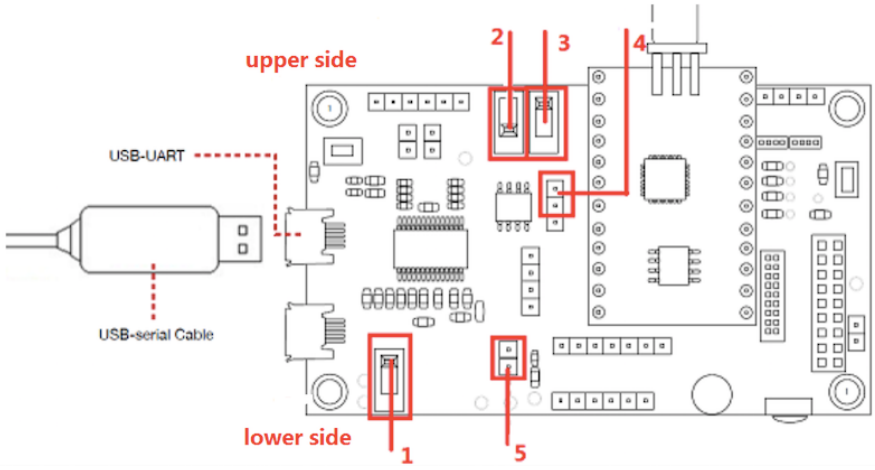


Figure II-3. ESP-Launcher

   - 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.

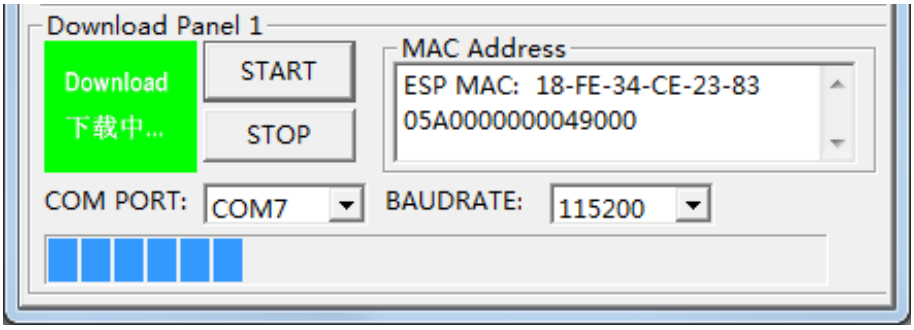


Figure II-4. Starting Downloading Firmware

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

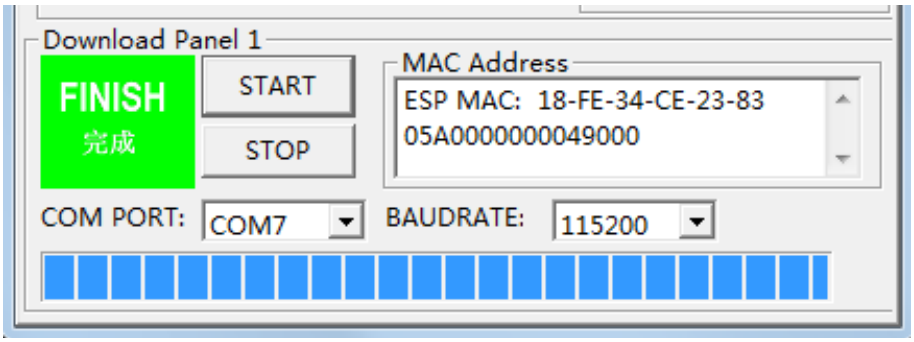


Figure II-5. Downloading Firmware Finished

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

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

III. Appendix - User Guide of Python Script

1. The ammeter used in this demonstration is an Agilent 34401A, which connects to the PC through the GPIB interface.
2. Pyvisa needs to be installed to run the Python script used in this demo.
3. After all software components and dependencies have been installed, connect the ammeter USB to the PC. For more on how to connect the multimeter, refer to its user manual.
4. Download the firmware into the ESP8266 demo board using the Flash Download Tool. Then run the firmware with baud rate set to 74880.
5. Adjust the ammeter to measure the current of ESP8266 working normally (in active mode). Then set the ESP8266 into different low-power modes, and run the python script every time to measure the current of ESP8266 and draw a conclusion about the average current.

Espressif IOT Team

www.espressif.com