# ESP8266 Quick Start Guide



Version 1.4 Copyright © 2017

# **About This Guide**

This document is a quick start guide to ESP8266. The document is structured as follows.

Chapter	Title	Content
Chapter 1	Configuring the Development Board ESP-LAUNCHER	Introduction to the ESP8266 development board ESP- LAUNCHER, and how to download firmware to the board and run it.
Chapter 2	Compiling ESP8266_NONOS_SDK	Compiling the ESP8266_NONOS_SDK in the virtual machine.
Chapter 3	Compiling ESP8266_RTOS_SDK	Compiling the ESP8266_RTOS_SDK.
Chapter 4	Common Debug Methods	Common debugging methods and sample codes.
Chapter 5	Downloading Firmware into the ESP-WROOM-02	Instructions on how to download firmware with ESP- WROOM-02.
Appendix A	Learning Resources	ESP8266-related must-read documents and must-have resources.

## **Release Notes**

Date	Version	Release notes
2016.08	V1.0	First release.
2016.11	V1.1	Added Appendix A-Learning Resources.
2017.01	V1.2	Added two Github links to RTOS and non-OS SDK sample code in Appendix A.2—Must-Have Resources.
2017.02	V1.3	Updated the link to the OVA image file in Section 2.1.
2017.05	V1.4	Updated Chapter 1.

# **Table of Contents**

1.	Conf	iguring the Development Board ESP-LAUNCHER1
	1.1.	Hardware Preparation1
	1.2.	Software Preparation1
	1.3.	Downloading Firmware into the ESP-LAUNCHER2
2.	Com	piling ESP8266_NONOS_SDK7
	2.1.	Downloading the Virtual Machine7
	2.2.	Setting up Development Environment
	2.3.	Compiling Process
3.	Com	piling ESP8266_RTOS_SDK12
	3.1.	Compiling Process
	3.2.	ESP8266_RTOS_SDK Architecture15
4.	Com	mon Debug Methods16
	4.1.	Common Debug Methods16
		4.1.1. Add UART Output Logs16
		4.1.2. Debug Fatal Exception
5.	Dow	nloading Firmware into the ESP-WROOM-0217
A.	Арре	endix – Learning Resources19
	A.1.	Must-Read Documents
	A.2.	Must-Have Resources



# 1. Configuring the Development Board ESP-LAUNCHER

# 1.1. Hardware Preparation

To start developing applications for the ESP8266, users need the following hardware and its corresponding software tools:

- One of the ESP8266 Hardware Development Kits (HDKs) specified below:
  - ESP8266's development board, ESP-LAUNCHER, as shown in Table 1-1.
  - ESP8266's module, ESP-WROOM-02, as described in Chapter 5.
- PC for programming: Windows XP or Windows 7 OS is recommended, with enough RAM to run a Linux virtual machine.
- Micro-USB cable.

#### Dotes:

- When users deploy third-party development boards or modules that integrate ESP8266, they should use development firmware provided by the corresponding manufacturers.
- To purchase ESP-WROOM-02 or ESP-LAUNCHER, please visit Espressif's official online store at: https://espressif.taobao.com,

## Table 1-1. The ESP-LAUNCHER

- 1 ESP-LAUNCHER
- 1 Micro-USB cable



## Notice:

The ESP8266 Wi-Fi module requires 3.3V power supply capable of delivering a peak current of 500 mA.

## 1.2. Software Preparation

- ESP8266's Flash Download Tool
  - Download it from: ESP8266 Flash Download Tools
- ESP8266's SDK
  - Download the SDK from: <u>ESP8266 SDK</u>



- The official AT firmware (*ESP8266\_NONOS\_SDK\bin\at*) can be downloaded into the ESP-LAUNCHER by referring to the BIN locations included in the *Readme* file which is in the same directory. For instructions on downloading the firmware into ESP-LAUNCHER, please refer to *Section 1.3*.
- UART terminal emulator tool

The default baud rate of ESP8266 is74880, therefore, UART tools that can support the default baud rate are recommended. Please note that certain USB-UART converters may not support all baud rates if users use a third-party development board.

# 1.3. Downloading Firmware into the ESP-LAUNCHER

 We use the ESP8266\_NONOS\_SDK\_V2.0.0\_16\_07\_19 as an example. The AT firmware binaries are located in ESP8266 NONOS SDK V2.0.0 16 07 19\ESP8266 NONOS SDK\bin.



Figure 1-1. ESP8266\_NONOS\_SDK BIN Folder

2. Settings of the development board, ESP-LAUNCHER.



Figure 1-2. The ESP-LAUNCHER

• switch 1: toggle to the lower side;



- switch 2: toggle to the lower side;
- *switch 3*: toggle to the upper side;
- *pin4*: put a jumper cap on the two pins above;
- *pin5*: put a jumper cap on it.
- 3. Use a Micro-USB cable to connect ESP-LAUNCHER to the PC. The UART driver needs to be installed on the PC.
- 4. Double-click on *ESPFlashDownloadTool\_v3.3.4.exe* to run the ESP8266 Flash Download Tool on the PC.

■ ESP8266 DOWNLOAD TOOL V3.3.4 - □ ×					
SPIDownlo	ad	HSPID	ownload	RFConfig	MultiDownload
Download P	ath C	Config			
✓ _07_19	ESP8	266_N	ONOS_SE	K\bin\blank.b	oin @ 0x1FB000
☑ 6_NON	os_s	DK\bin	\esp_init_	data_default.b	oin @ 0x1FC000
✓ _07_19	ESP8	3266_N	ONOS_SE	K\bin\blank.b	oin @ 0xFE000
√ _07_19	ESP8	8266_N	ONOS_SE	K\bin\blank.b	oin @ 0x1FE000
. 19\ESP	8266	NONC	DS_SDK\b	in\boot_v1.6.b	oin @ 0x00000
☑ SDK\bin	n∖at∖1	024+1	024\user1	1.2048.new.5.b	oin @ 0x01000
DeviceMast	erKey	y Folde	r Path		
					@
SpiFlashCon CrystalFreq : 26M ▼ SPI SPEED € 40MHz C 26.7MHz C 20MHz C 80MHz	tig	Com De SPI N © QI © QC © DI © DC	bineBin fault MODE O DUT O DUT	FLASH SIZE 4Mbit 8Mbit 16Mbit 32Mbit 16Mbit-C 32Mbit-C	DoNotChgBin
Download Panel 1 IDLE 等待 COM: COM9 BAUD: 1152000					

Figure 1-3. ESP8266 Flash Download Tool

Figure 1-3 uses **16Mbit-C1** (1024+1024 map) flash as an example. The download addresses are shown in Table 1-2.



#### Table 1-2. Download AT Binaries for 16 Mbit-C1 Flash Map

BIN	Address	Description
blank.bin	0x1FB000	It initializes the RF_CAL parameter area.
oon init data dafault hin	0x1FC000	It stores the default RF parameter values and has to be downloaded into the flash at least once.
esp_mi_data_derauit.pm		If the RF_CAL parameter area is initialized, this BIN has to be downloaded too.
blank.bin	0xFE000	It initializes the user parameter area.
blank.bin	0x1FE000	It initializes the system parameter area.
boot.bin	0x00000	It is the main program located in \bin\at.
user1.2048.new.5.bin	0x01000	It is the main program located in \bin\at\1024+1024.

#### Note:

The **SpiFlashConfig** and the **COM** on the ESP8266 Flash Download Tool should be set according to the actual situation.

- 5. Click the START button and wait for the ESP-LAUNCHER to power up.
- Power on the ESP-LAUNCHER by toggling *switch 1* as is shown in Figure 1-2 to the upper side. Toggle *switch 2* as is shown in Figure 1-2 to the lower side to enter the download mode.
- 7. The ESP8266 Flash Download Tool will start to download AT firmware into the ESP-LAUNCHER. The *DETECTED INFO* area will display the information of the flash chip on ESP-LAUNCHER.



SpiFlashConfig			
CrystalFreq :	CombineBin	FLASH SIZE	SpiAutoSet
26M 💌	Default	C 4Mbit	DoNotChgBin
SPI SPEED	SPI MODE	C 2Mbit	LOCK SETTINGS
④ 40MHz	QIO	C 8Mbit	DETECTED INFO
C 26.7MHz	C QOUT	C 16Mbit	flash vendor:
C 20MHz	C DIO	C 32Mbit	flash devID:
C 80MHz	C DOUT	I6Mbit-C1	4016h
		C 32Mbit-C1	Crystal:
			2 Mhz +
- Download Panel 1			
Download AP	MAC: 5E-CF-7F	-14-C7-45	
下载中 STA MAC: 5C-CF-7F-14-C7-45			
START STOP COM: COM9			
BAUD: 1152000			

Figure 1-4. ESP8266 Flash Download Tool - Downloading Firmware

8. After the download is finished, as Figure 1-5 shows, toggle *switch 1* to the lower side to power off the ESP-LAUNCHER.

Download Panel 1			
FINISH 完成	AP MAC: 5E-CF-7F-14-C7-45 STA MAC: 5C-CF-7F-14-C7-45		
START	STOP	COM:	COM9
L		BAUD:	1152000

Figure 1-5. ESP8266 Flash Download Tool–Finishing Downloading Firmware

9. Open the UART tool on the PC, set the baud rate to 115200, and configure new line mode.

## Dote:

When using AT commands, then a baud rate of 115200 and new line mode are needed.

10.Set the ESP8266 to working mode by toggling *switch 2* to the upper side. Then toggle *switch 1* to the upper side to power on the ESP-LAUNCHER.



The log print may look garbled, as shown in Figure 1-6 (which is normal, because the default power-on baud rate of ESP8266 is 74880). The "ready" message indicates that ESP-LAUNCHER is working successfully.

11.Input command AT+GMR and press *Enter* button. The information of the AT firmware will appear, as Figure 1-7 shows.

```
|嫩r?b呙ff?€frfjrfju[鄭fj?霉1滥?b咪ff寣b嗄
f€j+b咪ff?|?rfjrfj
AT+GMR
AT version:1.3.0.0(Jul 14 2016 18:54:01)
SDK version:2.0.0(656edbf)
compile time:Jul 19 2016 18:44:22
OK
```

Figure 1-6. AT Logs on the UART Tool

For more AT commands and examples, please refer to documents: <u>ESP8266 AT Instruction Set</u> and <u>ESP8266 AT Command Examples</u>.



# Compiling ESP8266\_NONOS\_SDK

This chapter describes how to compile *ESP8266\_NONOS\_SDK* by using the AT demo as an example.

# 2.1. Downloading the Virtual Machine

- 1. PC: Windows XP or Windows 7 OS is recommended.
- 2. The development environment provided by Espressif Systems is based on Lubuntu. We also provide a virtual image of the development environment that can be run on VirtualBox.
  - Download VirtualBox-5.0.16-105871-Win.exe from: https://www.virtualbox.org/wiki/Downloads
  - Download ESP8266\_lubuntu\_20141021.ova from: <u>http://downloads.espressif.com/FB/ESP8266\_GCC.zip</u>

# 2.2. Setting up Development Environment





Steps	Results
<ul> <li>Create a new folder, for example, <i>D</i>: \VM.</li> <li>Select <i>File &gt; Preferences</i>, and the system will show the dialog box </li> <li>In the <i>General</i> tab, set the location for the virtual machine in <i>Default Machine Folder</i>, for example, <i>D</i>: \VM.</li> <li>Tip:</li> <li>ESP8266's virtual machine takes up much space (RAM). Please make sure the machine has enough memory to</li> </ul>	VitualBox - Settings          General       ?         Input       General         Update       Default Machine Folder:         Language       Display         Network       Extensions         Proxy       VBoxAuth
spare.	OK Cancel Help
<ul> <li>Select <i>File</i> &gt; <i>Import Appliance</i>, and a dialog box will show up </li> <li>Select the image file, for example, <i>C</i>: \<i>ESP8266_lubuntu_20141021.ova</i>, and click on <i>Next</i>.</li> <li>Click <i>Import</i> to confirm the settings.</li> </ul>	Import Virtual Appliance         Appliance to import         VirtualBox currently supports importing appliances saved in the Open Virtualization Format (OVF). To continue, select the file to import below.         C:\ESP8266_lubuntu_20141021.ova         Expert Mode       Next         Cancel
4. Create a shared folder.	
<ul> <li>Create a new folder named <i>D:</i> \<i>VM\share</i>.</li> <li>Select <i>Machine</i> &gt; <i>Settings</i> &gt; <i>Shared Folders</i>, and a dialog box will show up </li> <li>Select the shared folder in <i>Machine</i> <i>Folders</i>, for example, <i>D:\VM\share</i>.</li> </ul>	SP8266_lubuntu_1 - Settings   General Shared Folders   System Folders List   Display Storage   Audio Machine Folders   Audio Machine Folders   Name Path   Audio Serial Ports   USB Shared Folders
	OK Cancel Help





# 2.3. Compiling Process

- 1. Start the virtual machine. Run *LXTerminal* on the desktop.
- 2. Copy the **ESP8266\_NONOS\_SDK** to the shared folder.



Steps	Result
<ul> <li>Copy the <i>ESP8266_NONOS_SDK</i> folder to the shared directory, for example, <i>D:\VM\share</i>.</li> <li>Copy the <i>ESP8266_NONOS_SDK\examples\at</i> folder to the directory <i>D:</i> \<i>VM\share\ESP8266_NONOS_SDK</i>, as shown in the figure on the right <i><sup>-</sup></i>.</li> </ul>	VM > share > ESP8266_NONOS_SDK at bin documents driver_lib examples include Id ib tools

3. Mount the shared directory to the virtual machine.

	Steps	Result
•	Execute command ./mount.sh in the LXTerminal. Input the password <i>espressif</i> . Shared folder mounting is completed.	<pre>esp8266@esp8266-VirtualBox:~ - + × File Edit Tabs Help esp8266@esp8266-VirtualBox:~\$ ./mount.sh [sudo] password for esp8266: esp8266@esp8266-VirtualBox:~\$ [</pre>
•	<ul> <li>Open the shared directory</li> <li><i>ESP8266_NONOS_SDK</i> in the virtual machine and check if the mounting has been successful.</li> <li>If successful, the directory contains files as the figure on the right  shows.</li> <li>If not, the directory will be empty and users will need to repeat the previous step.</li> </ul>	ESP8266_NONOS_SDK       - + ×         File Edit View Bookmarks Go Tools Help         File Edit View Bookmarks Go Tools Help         Fl < < > <          Places         Chome Folder         Desktop         Trash Can         Chome Folder         Documents         Music         Pictures         Videos         Tools         License         Makefile         11 items (1 hiden)



4. Change the directory to /Share/ESP8266\_NONOS\_SDK/at in the LXTerminal.

	Steps	Result
•	In the LXTerminal, execute the command: cd /home/esp8266/Share/ESP8266_NONOS_SDK/at	<pre>esp8266@esp8266-VirP8266_NONOS_SDK/at - + × File Edit Tabs Help esp8266@esp8266-VirtualBox:~\$ ./mount.sh [sudo] password for esp8266: esp8266@esp8266-VirtualBox:~\$ cd /home/esp8266/Share/ESP8 266_NONOS_SDK/at</pre>
•	Then execute command: ./gen_misc.sh to start compiling.	esp8266@esp8266-VirtualBox:~/Share/ESP8266_NONOS_SDK/at\$ ./gen_misc.sh gen_misc.sh version 20150511 Please follow below steps(1-5) to generate specific bin(s
•	For example, the options for STEP 1 ~ 5 can be: 1, 1, 2, 0, 5.	): STEP 1: choose boot version(0=boot_v1.1, 1=boot_v1.2+, 2= none) enter(0/1/2, default 2):

#### Dote:

For more details on compiling, please refer to ESP8266 SDK Getting Started Guide.

5. After compilation, the binaries generated and their corresponding download addresses in the flash memory will be shown as follows:

Support boot\_v1.4 and +

Generate user1.2048.new.5.bin successfully in folder bin/upgrade.

boot.bin---->0x00000

user1.2048.new.5.bin--->0x01000

!!!

#### Dote:

Users can open the *home/esp8266/Share/ESP8266\_NONOS\_SDK/bin/upgrade* directory and check the binaries compiled.

 The AT binaries generated can be downloaded to the ESP-LAUNCHER (refer to Section 1.3) and run.



3.

# Compiling ESP8266\_RTOS\_SDK

# 3.1. Compiling *Process*

1. Download *ESP8266\_RTOS\_SDK* from:

https://github.com/espressif/ESP8266\_RTOS\_SDK

ESP8266\_IOT\_PLATFORM is a demo application based on ESP8266\_RTOS\_SDK.

Download from: <a href="https://github.com/espressif/ESP8266\_IOT\_PLATFORM">https://github.com/espressif/ESP8266\_IOT\_PLATFORM</a>

2. Copy ESP8266\_IOT\_PLATFORM and ESP8266\_RTOS\_SDK to the shared folder.

Steps	Result
Copy <b>ESP8266_RTOS_SDK</b> and <b>ESP8266_IOT_PLATFORM</b> to the shared folder,	$\rightarrow$ VM $\rightarrow$ share
for example, <i>D:\VM\share</i> as shown on the right	
	ESP8266_IOT_PLATFORM

3. Start the virtual machine. Run *LXTerminal*.

	Steps	Result
•	<ul> <li>Start ESP8266 virtual machine  .</li> <li>If the virtual machine is locked, please enter the password: <i>espressif</i> to unlock it.</li> </ul>	ESP8266_lubuntu (Running) - Oracle VM VirtualBox — — X Machine View Devices Help
•	Double-click on <i>LXTerminal</i> to start the compilation.	LXTerminal



4. Mount the shared directory to the virtual machine.

	Steps	Result
•	Execute command ./mount.sh in the LXTerminal. Input the password: <i>espressif</i> . Shared folder mounting is completed.	<pre>esp8266@esp8266-VirtualBox:~ - + × File Edit Tabs Help esp8266@esp8266-VirtualBox:~\$ ./mount.sh [sudo] password for esp8266: esp8266@esp8266-VirtualBox:~\$ []</pre>
•	<ul> <li>Open the shared directory</li> <li><i>ESP8266_RTOS_SDK</i> in the virtual machine and confirm if the mounting has been successful.</li> <li>If successful, the directory contains files as shown in the figure on the right </li> <li>If not, the directory will be empty, and users will need to repeat the previous step.</li> </ul>	ESP8266_RTOS_SDK       - + ×         File Edit View Bookmarks Go Tools Help         It come Folder         Desktop         In Come Folder         Documents         In Applications         In Nusic         Include         Include         It is         Downloads         License         Videos         Include         Include         It is         Makefile         README.md         Prespace: 192.7 GIB (Total: 394.4 GB)

5. Modify the file *ESP8266\_IOT\_PLATFORM/gen\_misc.sh*. Set the variable PATH to point to the SDK and binaries.

Steps		Result	
	• SDK_PATH: the path of ESP8266_RTOS_SDK	🔚 gen_miso. sh 🗵	
	• <i>BIN_PATH</i> : the path where the generated binaries are stored after compilation	10 11 export SDK_PATH=~/Share/ESP8266_RTOS_SDK 12 export BIN_PATH=~/Share/ESP8266_RTOS_SDK/bin	
	<ul> <li>Please see the figure on the right 4.</li> </ul>		

6. Modify ESP8266\_IOT\_PLATFORM/makefile.

Steps	Result
Modify the LINKFLAGS_eagle.app.v6 area in the <b>ESP8266_IOT_PLATFORM/makefile</b> by: - Deleting -lminic - Adding -lcirom and -lmirom	LINKFLAGS_eagle.app.v6 = \



7. Switch to /Share/ESP8266\_IOT\_PLATFORM in the LXTerminal.

Steps	Result
<ul> <li>Execute the command: cd /home/esp8266/Share/ESP8266_IOT_PLATFORM</li> <li>Then execute the command: ./gen_misc.sh to start compiling.</li> </ul>	<pre>esp8266@esp8266-VirtualBox:hare/ESP8266_IOT_PLATFORM - + × File Edit Tabs Help esp8266@esp8266-VirtualBox:-\$ cd /home/esp82666/Share/ESP8266_IOT_PLATFORM esp8266@esp8266-VirtualBox:-/Share/ESP8266_IOT_PLATFORM\$ ./gen_misc.sh gen_misc.sh version 20150911 SDK PATH: /home/esp8266/ESP8266_RTOS_SDK BIN_PATH: /home/esp8266/ESP8266_RTOS_SDK/bin Please check SDK_PATH &amp; BIN_PATH, enter (Y/y) to continue: </pre>

## Dote:

For more details on compilation, please refer to ESP8266 SDK Getting Started Guide.

8. After compilation, the binaries generated and their download addresses in the flash memory will be shown as follows:

```
Support boot_v1.4 and +
Generate user1.1024.new.2.bin successfully in BIN_PATH
boot.bin---->0x00000
user1.1024.new.2.bin--->0x01000
!!!!
```

#### Dote:

Users can open the *lhome/esp8266/Share/ESP8266\_RTOS\_SDK/bin* directory and check the binaries compiled.

9. Download the generated binaries to the ESP-LAUNCHER and run.

#### I Note:

The default baud rate of ESP8266 is 74880.

# 3.2. ESP8266\_RTOS\_SDK Architecture



Figure 3-1. ESP8266\_RTOS\_SDK Architecture



# 4. Common Debug Methods

# 4.1. Common Debug Methods

## 4.1.1. Add UART Output Logs

For *ESP8266\_NONOS\_SDK*, users can add debug logs as below:

os\_printf("SDK version:%s\n", system\_get\_sdk\_version());

For *ESP8266\_RTOS\_SDK*, users can add debug logs as below:

printf("SDK version:%s\n", system\_get\_sdk\_version());

## 4.1.2. Debug Fatal Exception

If a fatal exception occurs, the UART output logs will be shown as below:

Fatal exception (28):

epc1=0x4025bfa6, epc2=0x00000000, epc3=0x00000000, excvaddr=0x00000000f, depc=0x000000000

The debugging steps are as follows:

1. Find the .s file that corresponds to the firmware that is currently running.

For example, if *eagle.flash.bin* and *eagle.irom0text.bin* are running, then the corresponding file is *eagle .s*.

- 2. Locate the address of epc1 (in the form of 0x40XXXXX) in the **.s** file to find the target function.
- 3. Add the UART logs before and after the target function to debug the fatal exception problem.



# 5. Downloading Firmware into the ESP-WROOM-02

Please follow the steps below to download firmware into ESP-WROOM-02.

1. ESP-WROOM-02 is the official ESP8266 module provided by Espressif Systems. Lead out the pins of ESP-WROOM-02, as shown in Table 5-1.

Pin	Pin status	Figure
EN	Pull up	
3V3	3.3V power supply (VDD)	
I015	Pull down	Pb All ava
100	UART Download mode: pull down; FLASH Boot mode: floating/pull up	TOTO IO16 EN IO14
GND	GND	105 CND
RXD	UART Download Rx	
TXD	UART Download Tx, floating/pull up	GND B

#### Table 5-1. ESP-WROOM-02 Pins

2. Connect ESP-WROOM-02 to the USB-to-TTL converter, using Dupont lines, as shown in Figure 5-1.



Figure 5-1. ESP-WROOM-02 Download Mode

- 3. Connect the USB-to-TTL converter to the PC.
- 4. Download firmware to flash with the ESP8266 Flash Download Tool.



## Dote:

On how to download firmware, please refer to Section 1.3.

- 5. After downloading, set **I00** as floating or pull up and switch ESP-WROOM-02 to working mode.
- 6. Power on ESP-LAUNCHER again and the chip will read and run programs from the flash.

## Dote:

**100** is an internal pull-up pin. For more information on the ESP-WROOM-02 hardware, please refer to ESP8266 System Description and ESP-WROOM-02 Datasheet.



# Appendix – Learning Resources

# A.1. Must-Read Documents

## ESP8266EX Datasheet

Description: This document introduces the specifications of ESP8266EX, including an overview of its features, protocols, technical parameters and applications. It also provides ESP8266EX's pin layout and the relevant description, as well as major functional modules and protocols applied on ESP8266EX (CPU, flash and memory, clock, radio, Wi-Fi, and low-power management). Furthermore, it provides a description of peripheral interfaces integrated on ESP8266EX, lists the electrical data of ESP8266EX and illustrates the package details for ESP8266EX.

• ESP8266 Hardware Resources

Description: This zip package includes the manufacturing specifications of the ESP8266 board and the modules, manufacturing bill of materials (BOM) and schematics.

<u>ESP8266 Non-OS SDK IoT\_Demo Guide</u>

Description: This document provides simple demo implementations of three types of smart devices: Smart Lights, Smart Power Plugs, and Sensor Devices. It also introduces the readers to curl toolkits, functions in LAN and WAN.

• ESP8266 RTOS SDK Programming Guide

Description: This document provides sample codes based on ESP8266\_RTOS\_SDK, including basic examples, networking protocol examples and advanced examples.

• ESP8266 AT Command Examples

Description: This document introduces some specific examples of using Espressif's AT commands, including setting up single connection using ESP8266 as a TCP client, setting up UDP transmission and transparent transmission, and setting up multiple connection using ESP8266 as a TCP server.

• ESP8266 AT Instruction Set

Description: This document provides lists of AT commands based on ESP8266\_NONOS\_SDK, including user-defined AT commands, basic AT commands, Wi-Fi AT commands and TCP/IP-related AT commands. It also introduces the downloading of AT firmware into flash.

• ESP8266 Non-OS SDK API Reference

Description: This document lists ESP8266\_NONOS\_SDK APIs, and provides an overview of the ESP8266\_NONOS\_SDK. It also introduces the readers to system APIs,



TCP/UDP APIs, mesh APIs, application-specific APIs, definitions and data structures, as well as APIs for peripheral interfacing.

- <u>ESP8266 RTOS SDK API Reference</u>
   Description: This document lists ESP8266\_RTOS\_SDK APIs, including functions for Wi-Fi-related APIs, and boot APIs, etc.
- <u>FAQ</u>

# A.2. Must-Have Resources

• ESP8266 SDKs

Description: This webpage provides links to the latest version of the ESP8266 SDK as well as the older ones.

<u>RTOS Sample Code</u>

Description: This webpage provides sample code for the commonly-used functions.

<u>Non-OS Sample Code</u>

Description: This webpage provides sample code for the commonly used functions.

• ESP8266 Tools

Description: This webpage provides links to the ESP8266 flash download tools and ESP8266 performance evaluation tools.

- <u>ESP8266 APK</u>
- ESP8266 Certification and Test Guide
- <u>ESP8266 BBS</u>
- ESP8266 Resources



# Espressif IOT Team www.espressif.com

#### **Disclaimer and Copyright Notice**

Information in this document, including URL references, is subject to change without notice.

THIS DOCUMENT IS PROVIDED AS IS WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR 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.

Copyright © 2017 Espressif Inc. All rights reserved.