Espressif
Systems

ESPRESSIF

ESP8266EX:

A BEGINNER'S GUIDE

CONFIDENTIAL

September 17, 2014

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, NONINFRINGEMENT, 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.

All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners, and are hereby acknowledged.

Copyright © 2013 Espressif Systems Inc. All rights reserved.

Table of Contents

Espressif
Systems

CONFIDENTIAL

Espressif
Systems

1. Introduction

1.1. General Overview

Espressif Systems' Smart Connectivity Platform (ESCP) is a set of high performance, high integration wireless SOCs, designed for space and power constrained mobile platform designers. It provides unsurpassed ability to embed Wi-Fi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minimal space requirement.
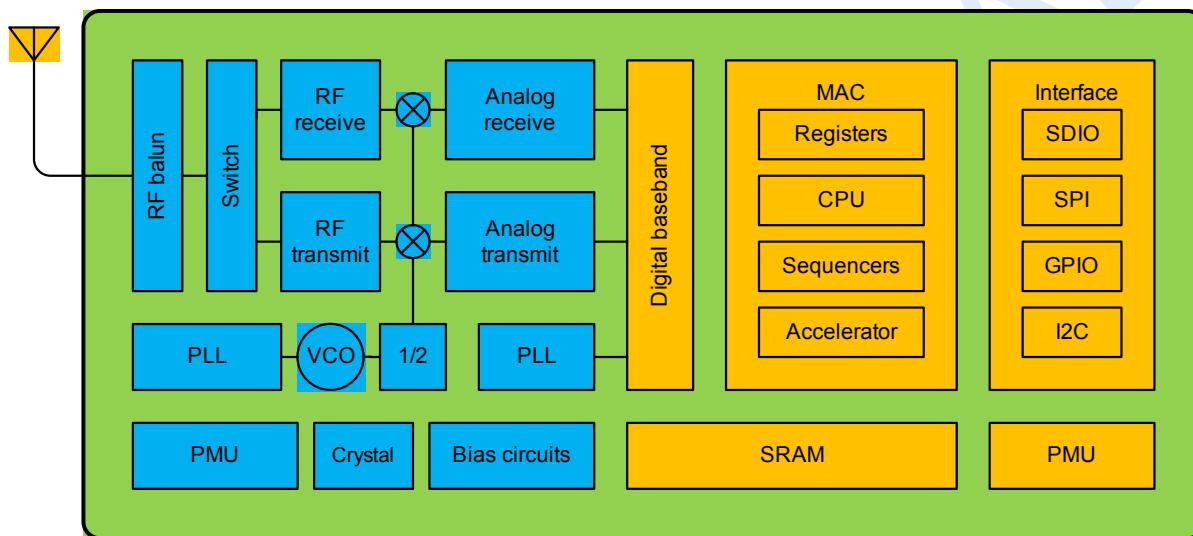
Figure 1: ESP8266EX Block Diagram

ESP8266EX offers a complete and self-contained Wi-Fi networking solution; it can be used to host the application or to offload Wi-Fi networking functions from another application processor.

When ESP8266EX hosts the application, it boots up directly from an external flash. In has integrated cache to improve the performance of the system in such applications.

Alternately, serving as a Wi-Fi adapter, wireless internet access can be added to any microcontroller-based design with simple connectivity (SPI/SDIO or I2C/UART interface).

ESP8266EX is among the most integrated WiFi chip in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power

management modules, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor, with on-chip SRAM, besides the Wi-Fi functionalities. ESP8266EX is often integrated with external sensors and other application specific devices through its GPIOs; codes for such applications are provided in examples in the SDK.

Sophisticated system-level features include fast sleep/wake context switching for energy-efficient VoIP, adaptive radio biasing for low-power operation, advance signal processing, and spur cancellation and radio co-existence features for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation.

1.1.1. Features

- 802.11 b/g/n protocol
- Wi-Fi 2.4 GHz, support WPA/WPA2
- Super small module size (11.5mm x 11.5mm)
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- +20dBm output power in 802.11b mode
- Supports antenna diversity
- Deep sleep power <10uA, Power down leakage current < 5uA
- Integrated low power 32-bit MCU
- SDIO 2.0, SPI, UART
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4µs guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- Operating temperature range -40C ~ 125C

1.1.2. Specifications

Category	Parameter	Value
Wi-Fi	Standard	CCC/FCC/CE
	Wi-Fi	802.11b/g/n
	Frequency	2.4G-2.5G (2400M-2483.5M)
	Tx Power	802.11b: 20dBm
		802.11g: 17dBm
		802.11n: 14dBm
	Rx Sensitivity	802.11b: (11Mbps) -91dBm
		802.11g: (54Mbps) -75dBm

Espressif
Systems

		802.11n: (MCS7) -72dBm
	Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip
Hardware	Data Interface	UART/SDIO/SPI/I2C
		PWM, GPIO
	Operating Voltage	3.0~3.6V
	Operating Current	Avg. 80mA
	Operating Temperature	-40 C ~ 125 C
	Storage Temperature	Room Temperature
	Size	5x5mm
	External Interface	N.A.
Software	Wi-Fi Mode	Station/SoftAP/SoftAP+Station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Flash Download / Via Cloud Server
	SW Development	Supports Cloud Server Development / SDK for custom firmware development
	Network Protocols	IPv4. TCP/UDP/HTTP/FTP
	User Config	AT Instruction Set, Cloud Server, Android/iOS App

September 17, 2014

1.1.3. Applications

- Smart power plugs
- Home automation
- Mesh network
- Industrial wireless control
- Baby monitors
- IP Cameras
- Sensor networks
- Wearable electronics
- Wi-Fi location-aware devices
- Security ID tags
- Wi-Fi position system beacons

September 17, 2014

1.2. Hardware Overview

1.2.1. Pin Definition
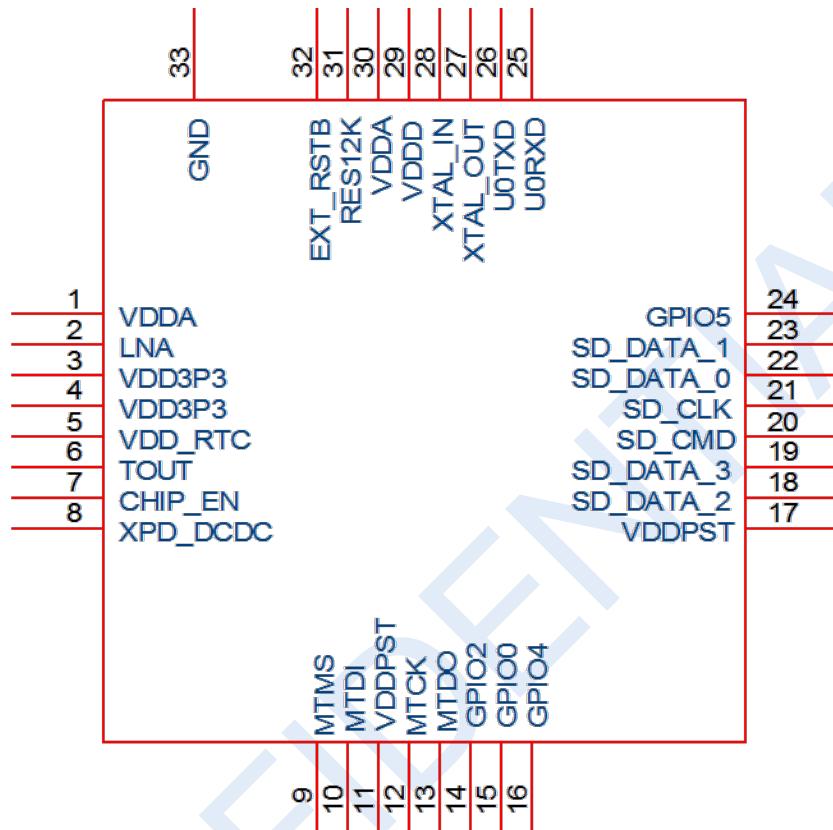
Figure 2: ESP8266EX Pin Definition Diagram

Pin	Name	Type	Function
1	VDDA	P	Analog Power 3.0 ~3.6V
2	LNA	I/O	RF Antenna Interface, Chip Output Impedance=50 Ω No matching required but we recommend that the π-type matching network is retained.
3	VDD3P3	P	Amplifier Power 3.0~3.6V
4	VDD3P3	P	Amplifier Power 3.0~3.6V
5	VDD_RTC	P	NC(1.1V)

6	TOUT	I	ADC Pin
7	CHIP_EN	I	Chip Enable. High: On, chip works properly; Low: Off, small current
8	XPD_DCDC	I/O	Deep-Sleep Wakeup；GPIO16
9	MTMS	I/O	GPIO14; HSPICLK
10	MTDI	I/O	GPIO12;HSPIQ
11	VDDPST	P	Digital/IO Power Supply (1.8V~3.3V)
12	MTCK	I/O	GPIO13;HSPID
13	MTDO	I/O	GPIO15;HSPICS
14	GPIO2	I/O	UART Tx during flash progamming；GPIO2
15	GPIO0	I/O	GPIO0;SPICS2
16	GPIO4	I/O	GPIO4
17	VDDPST	P	Digital/IO Power Supply (1.8V~3.3V)
18	SDIO_DATA_2	I/O	Connect to SD_D2 (Series R 200Ω);SPIHD; HSPIHD
19	SDIO_DATA_3	I/O	Connect to SD_D3 (Series R 200Ω); SPIWP; HSPIWP
20	SDIO_CMD	I/O	Connect to SD_CMD(Series R 200Ω); SPICS0
21	SDIO_CLK	I/O	Connect to SD_CLK (Series R 200Ω); SPICLK
22	SDIO_DATA_0	I/O	Connect to SD_D0 (Series R 200Ω); SPIQ
23	SDIO_DATA_1	I/O	Connect to SD_D1 (Series R 200Ω); SPID
24	GPIO5	I/O	GPIO5
25	U0RXD	I/O	UART Rx during flash progamming；GPIO3
26	U0TXD	I/O	UART Tx during flash progamming; GPIO1; SPICS1
27	XTAL_OUT	I/O	Connect to crystal output，can be used to provide BT clock input
28	XTAL_IN	I/O	Connect to crystal input
29	VDDD	P	Analog Power 3.0~3.6V
30	VDDA	P	Analog Power 3.0~3.6V
31	RES12K	I	Connect to series R 12kΩ to ground
32	EXT_RSTB	I	External reset signal (Low: Active)

Note: GPIO2, GPIO0, MTDO can be configurable as 3-bit SDIO mode

Espressif
Systems

1.2.2. Electrical Characteristics

Parameter	Condition	Min	Typical	Max	Unit
Storage Temperature		-40	Room	125	℃
Maximum Soldering Temperature	IPC/JEDEC J-STD-020			260	℃
Operating Voltage		3.0	3.3	3.6	V
Any I/O Pin Voltage	V_{IL}	-0.3/0.75V O		0.25V	V
	V_{OL}	N/0.8V		0.1V	V
	I_{MAX}			12	mA
Electrostatic Discharge (HBM)	TAMB=25℃			2	KV
Electrostatic Discharge (CDM)	TAMB=25℃			0.5	KV

1.2.3. Power Consumption

Parameter	Typical Value	Unit
Tx802.11b, CCK 11Mbps, P OUT=+17dBm	170	mA
Tx802.11g, OFDM 54Mbps, P OUT =+15dBm	140	mA
Tx802.11n, MCS7, P OUT =+13dBm	120	mA
Rx802.11b, 1024 byte packet size, -80dBm	50	mA
Rx802.11g, 1024 byte packet size, -70dBm	56	mA
Rx802.11n, 1024 byte packet size, -65dBm	56	mA
Modem-Sleep	15	mA
Light-Sleep	0.9	mA
Deep-Sleep	10	uA
Off	5	uA

[1] Modem-Sleep requires the CPU to be working, as in PWM or I2S applications. According to 802.11 standards (like U-APSD), it saves power to shut down the Wi-Fi Modem circuit while

maintaining a Wi-Fi connection with no data transmission. E.g. in DTIM3, to maintain a sleep 300ms-wake 3ms cycle to receive AP's Beacon packages, the current is about 15mA

[2] During Light-Sleep, the CPU may be suspended in applications like Wi-Fi switch. Without data transmission, the Wi-Fi Modem circuit can be turned off and CPU suspended to save power according to the 802.11 standard (U-APSD). E.g. in DTIM3, to maintain a sleep 300ms-wake 3ms cycle to receive AP's Beacon packages, the current is about 0.9mA

[3] Deep-Sleep does not require Wi-Fi connection to be maintained. For application with long time lags between data transmission, e.g. a temperature sensor that checks the temperature every 100s, sleep 300s and waking up to connect to the AP (taking about 0.3~1s), the overall average current is less than 1mA.

1.2.4. Package Footprint
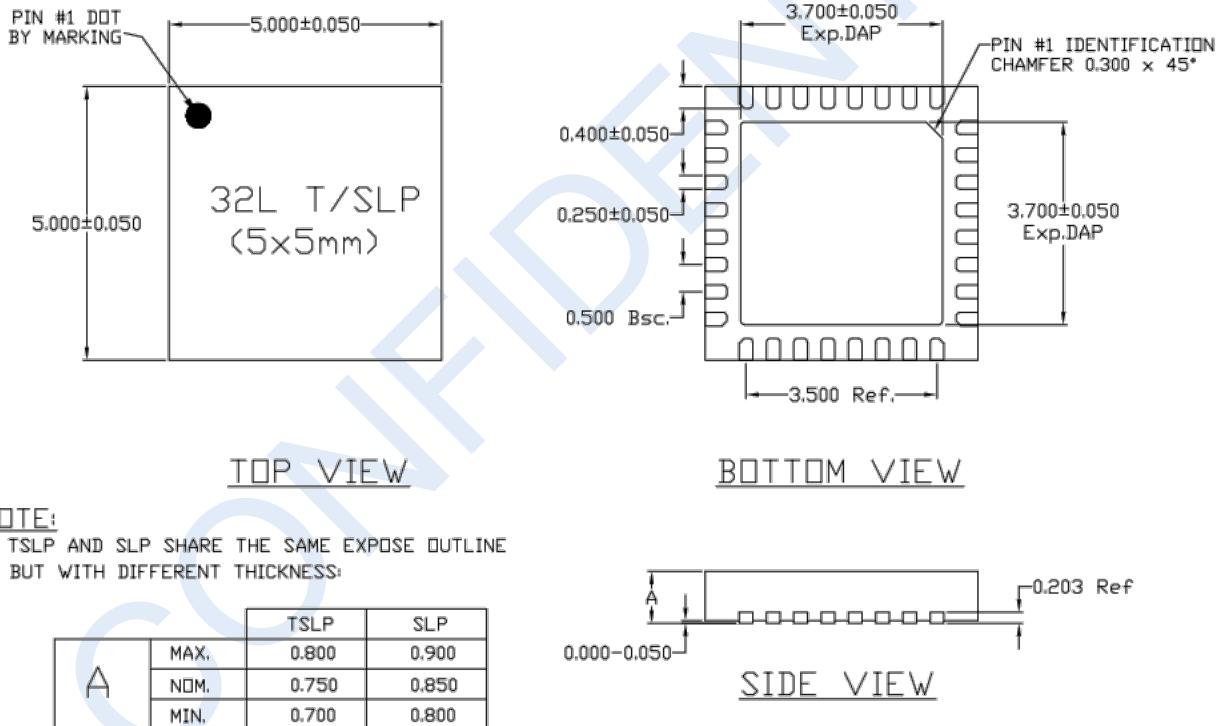
Figure 3: ESP8266EX Package Dimensions

MTDO	GPIO0	GPIO2	Mode
1	X	X	SDIO/SPI
0	0	1	UART Download
0	1	1	Flash Boot

1.2.5. Hardware Development Kit

Espressif provides a demo board for ESP8266EX.
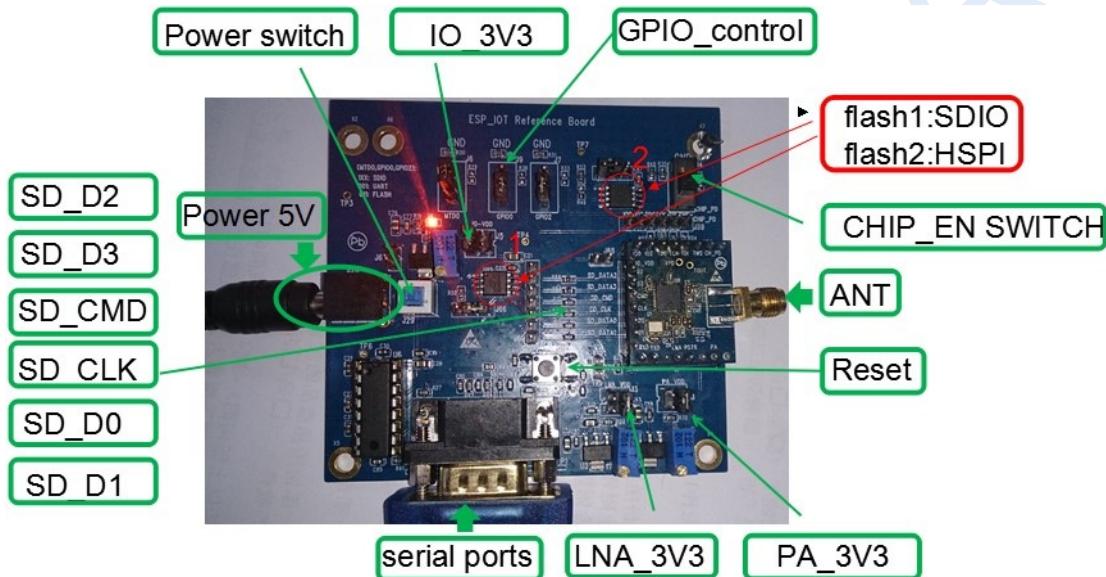
Figure 4: ESP8266EX Demo Board (Brief Outline)

- External power supply: 5V, 1A, positive in, negative out.

- External 5V power supply split through 3 LDO for ESP8266: PA, LNA and IO. Blue variable resistor can be used to adjust the value of the three power supplies. Please make sure the supply voltage is within the specified range.

- Reset button connected to EXT_RSTB, for external reset. Toggle switch is connected to the chip enable, please pay attention to maintaining a high (switch down) when used.

- Users can either use the RS-232 serial port or UART connector board for download, print log and serial communication.

- When connected to a typical external SMA antenna, RF performance tests can be done using cable lines connected to the test instrument.

- MTDO, GPIO0, GPIO2 form a 3-bit selectable SDIO mode. Please refer to board markings for jumper info. Confirm power mode settings are correct by printing log.

- Demo board has 2 SPI flash: flash1 and flash2 used as storage for firmware application development. Flash can be selected via jumper, CS high (H) to disable the flash.

  Flash1: Use SDIO connection, mostly used for standalone mode.

  Flash2: Use HSPI connection (multiplexing GPIO port). Mainly used in SIP mode, the SDIO (SPI) for external MCU, GPIO port multiplexing with HSPI to connect to flash.

1.2.6. ESP8266EX Modules (WROOM)

Espressif offers 2 kinds of modules, SMD and DIP.

SMD:
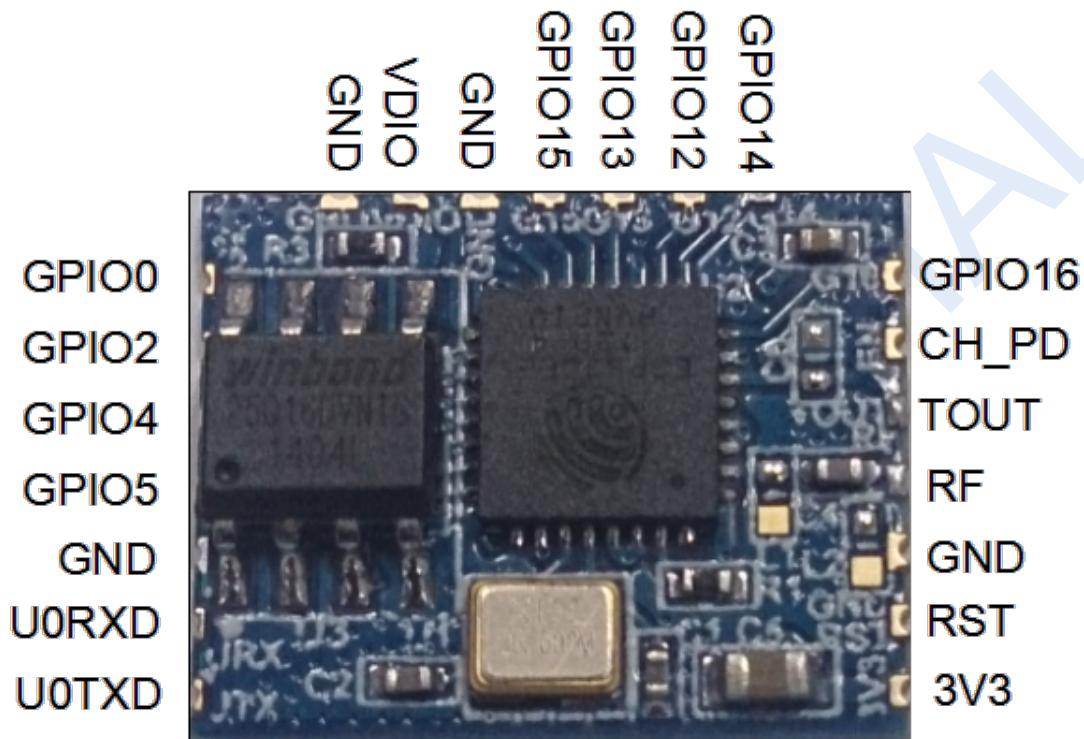
Figure 5: WROOM SMD Module

1. Refer to Figure 5 for pin definition. Module size is 11.8x15mm. Flash package is SOP8-150mil. Refer below for usage of SMD module (Tables in section 1.2.1 for reference):

2. Connect Pin 3V3 and VDIO to an external power source.

3. CH_PD: high.

4. MTDO: low, GPIO2 if not used may be left floating (high), GPIO0 to be switched between high and low for Uart Download and Flash Boot mode respectively.

5. Connect GND U0RXD U0TXD, use USB to TTL serial cable to download, print log and send data

DIP:

GND	GND
U0TXD	GND
U0RXD	GPIO5
RST	TOUT
CH_PD	GPIO16
GPIO4	GPIO0
GPIO2	GPIO15
GPIO13	GPIO12
VDD33	GPIO14
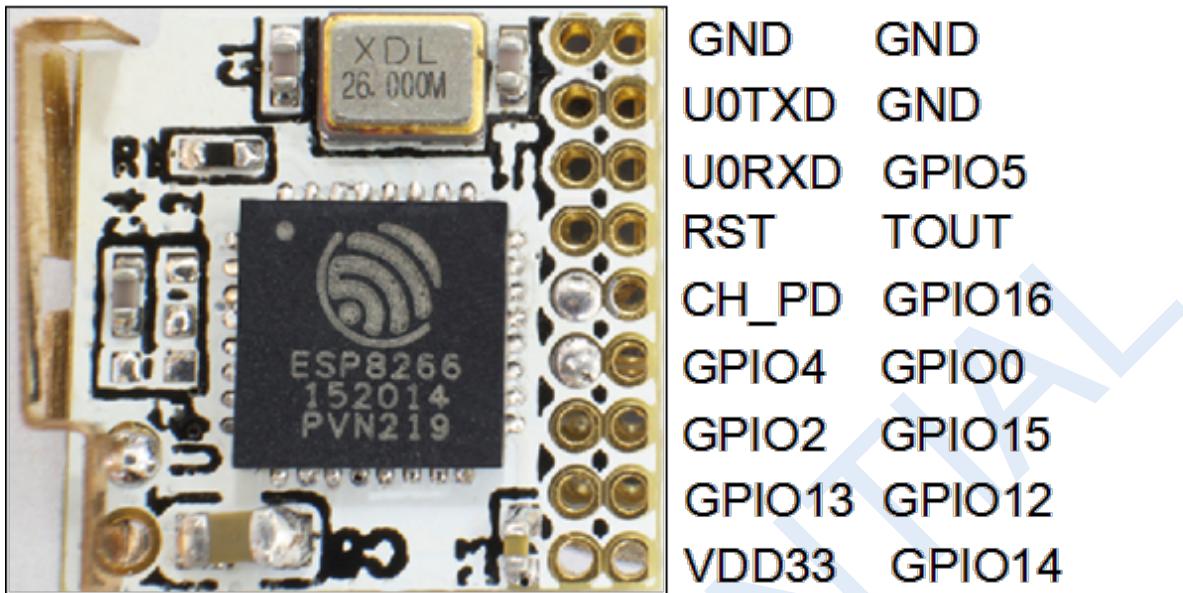
Figure 6: WROOM DIP Module

Refer to Figure 6 for and pin definition.

Features:

1. Ultra-small size: 11.5x11.5mm.

2. Flash package: USON8_2x3mm. Flash located on the back of the module.

3. External metal antenna. Module used in vertical position can reduce the interference of external circuit (antenna?).

4. The module can be plugged in directly, and is suitable for situations without a height limitation.

5. Refer to SMD module for usage.

1.3. Applications using ESP8266EX

1.3.1. UART Connector (as in Fig. 4 Demo Board)

PIN Definition for UART Connector:

UART0: (PIN 25) U0RXD+ (PIN 26) — Communication

UART1: (PIN 14) GPIO2(TXD) — Print Log

uart0 can be used to transmit and receive data packets while **uart1** can be used to print log.

Refer to AT commands section for usage.

Application: Used in demo board

1.3.2. Sensor Application (as in Fig. 5 USB Sensor Demo)

PIN Definition for Sensor Application:

(PIN 9)   MTMS — I2C_SCL

(PIN 14) GPIO2 — I2C_SDA

(PIN 12) MTCK — Reset button (Press on the reset button during power on)

(PIN 15) GPIO0 — Wi-Fi status indicator

(PIN 10) MTDI — Server communication status indicator

(PIN 25) U0RXD — Button (function to be defined)

(PIN 13) MTDO — LED: green light (function to be defined), used in smart plug demo relay control indicator light
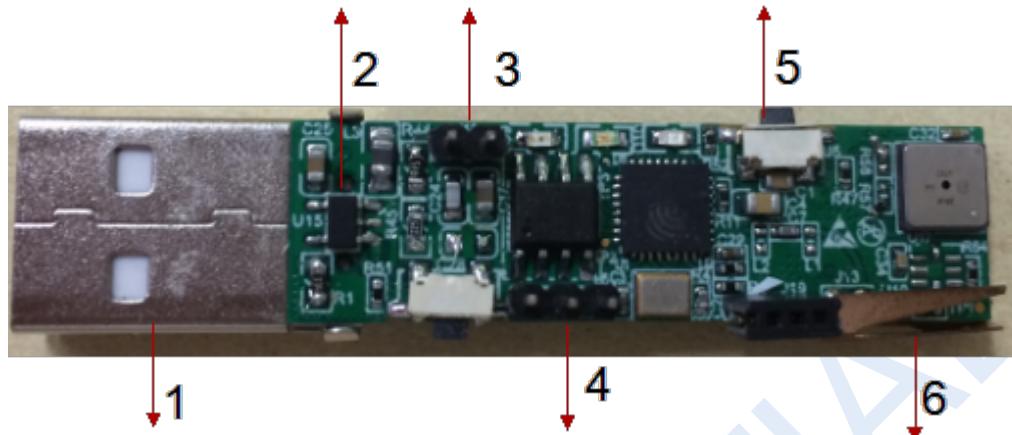
Figure 7: ESP8266EX as used in USB Sensor Demo

1. USB interface is used only for 5V power supply.

2. DC-DC power chip converts 5V input voltage into 3.3V for circuit use.

3. GPIO0 connector. With jumper cap: low; otherwise: high. For switching between UART Download and Flash Boot mode.

4. GND RXD TXD connector: connect to USB to TTL serial cable for download, print log.

5. Reset button. Hold down the reset button and power on to complete the reset.

6. External metal antenna.

1.3.3. Smart Light Application (as in Fig. 6 Smart Light Demo)

PIN Definition for Smart Light Application:

> (PIN 9) MTMS — Infrared receiver
>
> (PIN14) GPIO2 — Connect to reset button (Press for 5s to reset)
>
> Three PWM outputs:
>
> (PIN 10) MTDI — Red light control
>
> (PIN 13) MTDO — Green light control
>
> (PIN 12) MTCK — Blue light control
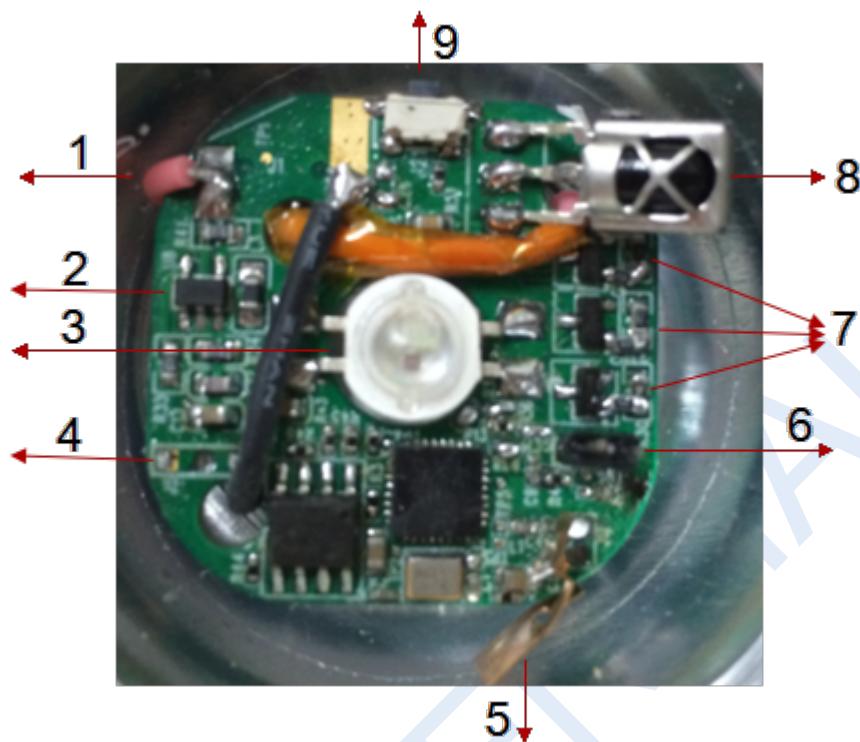
Figure 8: ESP8266EX as used in Smart Light Demo

1. Input voltage of 5V for RGB triple color light and other circuits.

2. 5V DC-DC power chip converts to 3.3V.

3. 3w RGB triple color light.

4. GND RXD TXD connector: connect USB to TTL serial cable to download, print log.

5. External metal antenna.

6. GPIO0 connector. With jumper cap: low; otherwise: high. For switching between UART Download and Flash Boot mode.

7. Three PWM outputs.

8. Infrared receiver transistor.

9. Reset button: Press 5s to complete reset.

1.3.4. Wi-Fi Smart Plug Application

PIN Definition for Wi-Fi Smart Plug Application:

(PIN 13) MTDO — Control relay

(PIN 15) GPIO0 — Wi-Fi status indicator

(PIN 10) MTDI — Communication (with server) indicator

(PIN 12) MTCK — Reset button (Hold for 5s to reset)

Sensors using ESP8266 (Figure 7) can be used to in the wifi smart plug demo. After the plug demo firmware is downloaded, the plug can be controlled by the client APP. The green light in the middle of the temperature sensor board is used to simulate relay control indicator (on and off).

2. Software Features

2.1. Wireless Networking

ESP8266EX supports 3 modes:

- SoftAP mode
- Station mode
- SoftAP + Station mode

Use ESP8266EX to achieve a flexible network topology.

2.1.1. SoftAP Mode

Using ESP8266EX in softAP mode will allow mobile phones, computers, user devices and other ESP8266EX station interfaces to connect to it, forming a LAN.

2.1.2. Station Mode

Using ESP8266EX in station mode will it to connect through routers (AP) to a Cloud server to upload and download data. The user can use apps on the mobile phones or tablets etc to monitor and control the ESP8266EX device.
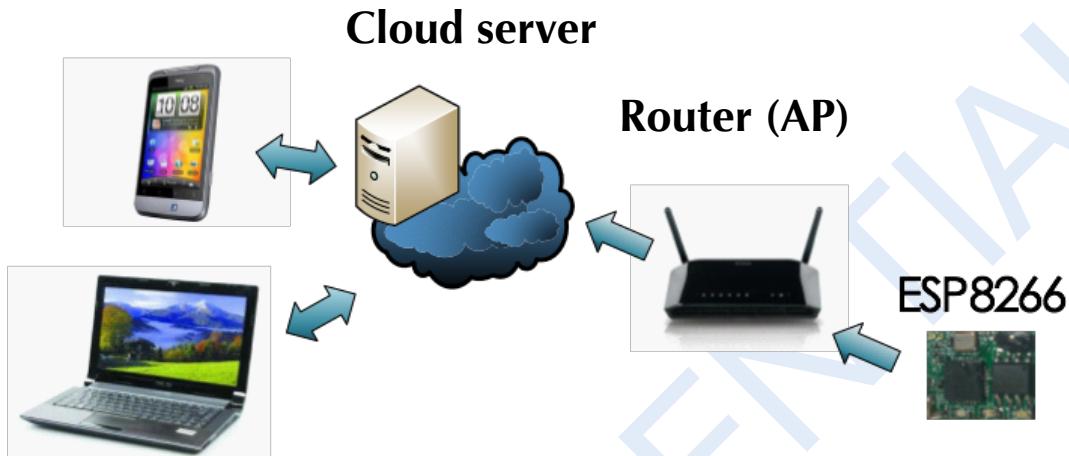
2.1.3. SoftAP + Station Mode

ESP8266EX can support softAP+station mode, and when user devices, mobiles etc can connect to ESP8266EX using softAP, they can also control it via the router to the Cloud server.
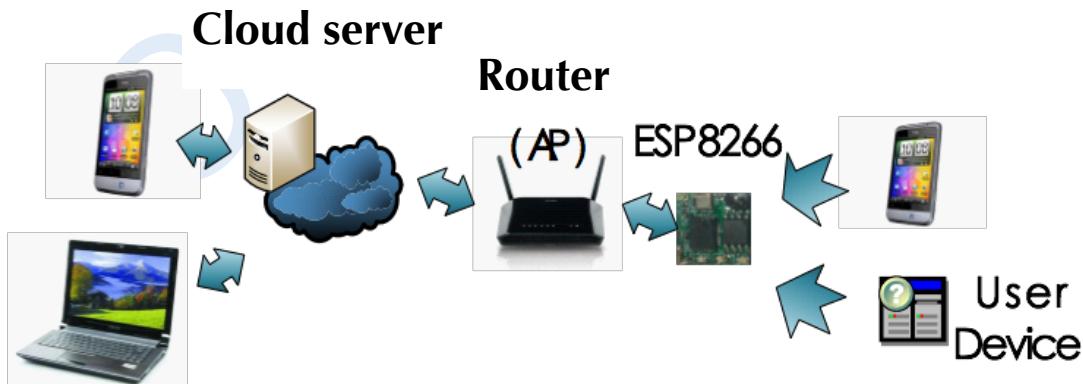
2.2. Pass-through Connection

Pass-through here refers to the transparent transmission function. Host data is transmitted to ESP8266EX through uart and ESP8266EX the transmits the data over the wireless network, and vice versa. ESP8266 receives data over a wireless network, empathy spread through uart Host. ESP8266 is only responsible for the data transmitted to the destination and it does not process the data in any way. The transmission process is as if it is transparent.

Parameters needed to establish a pass-through connection:	
Wireless network parameters	Network Name (SSID)
	Encryption
	Key (password)
TCP connection parameters	Protocol type
	Connection type (client)
	Destination IP Address
	Destination port
Serial port parameters	Baud Rate
	Data bits
	Check digit
	Stop bits
	Hardware flow control

2.3. UART Frames

ESP8266EX can estimate the interval between receiving UART data. If the interval is greater than 20ms, it is considered the end of one transmission packet. Otherwise, it will receive data to the upper limit of 2KB, and consider an end. When the ESP8266 module thinks that the UART data has reached the end of a frame, it will forward the data via Wi-Fi. Frame time interval is 20ms with a size limit of 2KB.

2.4. Encryption

ESP8266EX supports many different kinds of encryption:

- WEP (only in station mode)

- WPA-PSK/TKIP

- WPA-PSK/AES

- WPA2-PSK/TKIP

- WPA2-PSK/AES

Espressif
Systems

2.5. Low Power Operation

ESP8266EX supports 3 low power operation modes:

Mode	Modem-Sleep	Light-Sleep	Deep-Sleep
Action	Turn off Wi-Fi modem CPU and other peripherals are still running.	Turn off Wi-Fi modem, crystal and PLL CPU and other peripherals are suspended.	Only RTC circuit is working, all others are off, chip is in low power standby mode.
Current	10~20mA	0.5mA	10~20uA
Wake-up	Yes	Yes	Wake up based on defined interval settings only.
Usage	Used when CPU needs to be running all the time, e.g. PWM or I2S applications. If there is no data transmission, Wi-Fi modem can be turned off according to the 802.11 standards (U-APSD). E.g. During DTIM3, each cycle is 300ms, wake 3ms receive AP's Beacon packages, etc., overall average current of about 15mA.	Used in applications where the CPU can be suspended, e.g. Wi-Fi switch. If there is no data transmission, according to the 802.11 standard (such as U-APSD), turn off Wi-Fi modem and suspend CPU. E.g DTIM3, each sleep (300ms)—wake (3ms) receive AP's Beacon packages, etc., the overall average current of about 0.9mA. "	Used in applications where Wi-Fi need not be always connect, a long time before transmitting the application data packets. Measured once every 100 seconds, such as the temperature sensor. E.g. after waking up every 300S need 0.3 ~ 1s connected to the AP transmit data, the overall average current can be much less than 1mA.

2.6. Firmware Updater

In addition to the usual serial flash update, ESP8266EX also supports firmware update through Cloud server. Simply upload the new firmware to the Cloud server and when ESP8266EX is connected to the internet, the Cloud server will push the update to the user. The user can then choose to upgrade or not.

For serial flash update, refer to "Espressif IoT SDK Manual".

For Cloud update, refer to "Espressif Cloud Introduction"

3. Espressif Cloud Server

3.1. Guide to using Espressif Cloud Server Website

A. Goto http://iot.espressif.cn/#/

"Start 开始"->"Developer API 开发者API": API device control instructions

"Start 开始"->"Help 帮助": Example to help user setup their own products

"Register 注册": New user registration

"Login 登录": Login if registered user

B. After logging in, click on "Device Development 设备开发". User can edit the settings for all the devices.

"Search 搜索": Enter device name or device key to search for device

"Export 导出": Export the device list

"Create 创建": Create a new device
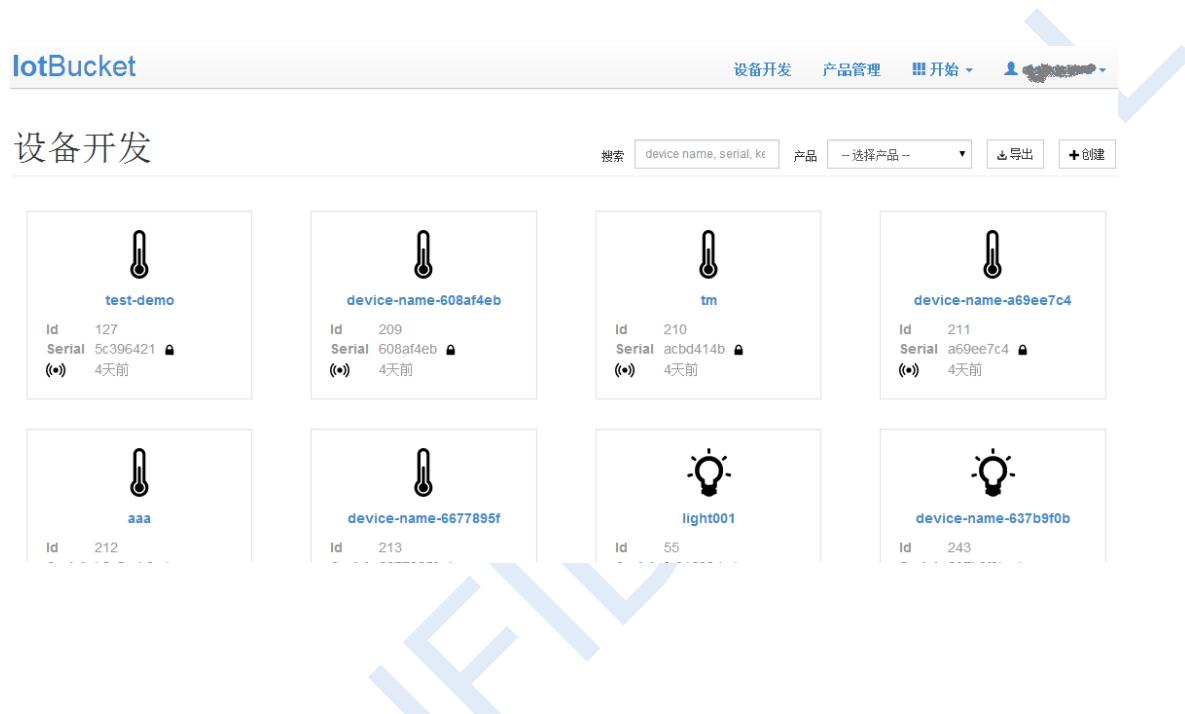
C. "Product Management 产品管理"shows the list of products. Use "Search 搜索", "Product 产品", "Status 状态" to filter and query products.

3.1.1. Device Development

In this menu, the user can view the device history, add timer, upgrade firmware and customize reverse control.

3.1.1.1. Device History

On the device page -> "Data Model 数据模型" -> "Data 数据" to view device history.

September 17, 2014

3.1.1.2. Debugging Info

"Request Log 请求日志" logs the commands received by the server and used to debug the device.

请求日志

POST /v1/ping/ 2014-06-10 12:24:35 ▼

POST /v1/datastreams/plug-status/datapoint/ 2014-06-10 12:22:49 ▼

POST /v1/datastreams/plug-status/datapoint/ 2014-06-10 12:22:43 ▼

POST /v1/ping/ 2014-06-10 12:22:39 ▼

POST /v1/ping/ 2014-06-10 12:21:44 ▼

POST /v1/ping/ 2014-06-10 12:20:52 ▼

3.1.1.3. Timer

Espressif Cloud server supports 3 kinds of timers:

Precise timer 精确时间: Execute a command at a specified time

Cycle Timer 周期循环: Execute a command at fixed intervals

Weekly Timer 按周循环: Execute a command at weekly or on certain days each week

Timer

在 20140709131520 执行动作 action1

每 4 hour 执行动作 action2

每周 [1,3,5]
在 161000 执行动作 action3

类型：

◉ 精确时间   ○ 周期循环   ○ 按周循环

时间	动作	+
20141011121314		

保存 保存到设备 取消

3.1.1.4. Customized Reverse Control

User can customize an action to reverse control the device.

RPC 请求

可以发送任意的 action 到设备，附带参数，首先选择请求密钥

device key 69██████████████████████████660db

请求参数 /v1/device/rpc/?deliver_to_device=true&

action=

请求

3.1.1.5. Firmware upgrade

User update the firmware through Cloud Server

ROM 发布

当前设备的 ROM 版本是，可以升级到 v1.1 ▼ 升级
v1.1
v1.0

3.1.2. Product Management

Here, the user can control the sales of products and upload firmware updates.

In the product list, the Activated/Total number shows how many products has been activated by the customers vs the total number of products made.

IotBucket

产品管理

Id	1
Name	humiture
Serial	f3ec8e37 🔒 (3天前)
Status	deployed
Description	description Of product(serial: f3ec8e37)
Activated / Total	86 / 552
	15.5797101 44927537%

Id	3
Name	light
Serial	9f4f24e2 🔒 (3天前)
Status	developing...

To upgrade a product's firmware, simply upload it and Espressif Cloud Server will push the update to all devices. The user can choose to upgrade or not.

ROM 发布

v1.1 当前版本
chore(beta): v1.1
🗎 user1.bin
🗎 user2.bin

v1.0 设置为当前版本
chore(beta): v1.0 codename(test)
🗎 user1.bin
🗎 user2.bin

September 17, 2014

3.2. Guide to using ESP8266EX modules

3.2.1. Software Debugging Tools

The following tools are recommended for debugging ESP8266EX modules. The user can also choose to use other similar tools.

- Flash programming tools: FLASH_DOWNLOAD_TOOLS.exe (provided in the SDK)

- Serial Transfer tools: SecureCRTPortable.exe

- Network Debugging tools: NetAssist.exe

3.2.2. Network Connections

ESP8266EX has 2 types of network interface, softAP and station. Both can be used at the same time. Depending on the user's actual requirements:

- SoftAP interface:

  Mobile phone or PC acts as station, connects to ESP8266EX through softAP interface. PC can connect to ESP8266EX serial port for debugging info.

- Station interface:

  ESP8266EX acts as station, connects to router (AP). PC can connect to ESP8266EX serial port for debugging info.

3.2.3. Default Connection Parameters

- Default SSID: ESP_XXXXXX

  (XXXXXX are the last 6 characters of module's MAC address)

- Default Encryption: WPA/WPA2

- Default Serial Parameters: 74880, 8, 1, None

- Default Mode: SoftAP

- Default IP Address: 192.168.4.1

3.3. Application Examples

3.3.1. Wi-Fi Remote Control

Example: Mobile phone acts as station, connects to ESP8266EX through softAP. ESP8266EX can connect to device via UART and mobile phone can now control the device.
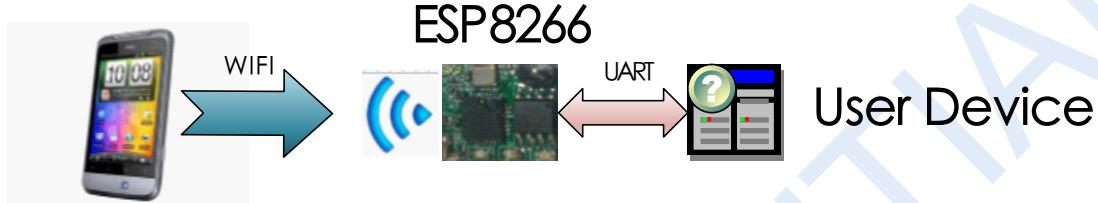
3.3.2. Wi-Fi Remote Access

Example: ESP8266EX acts as station, connects through router (AP) to internet. It can then upload or download data via Cloud server. Mobile phone/tablets etc can now control ESP8266EX via Cloud server.
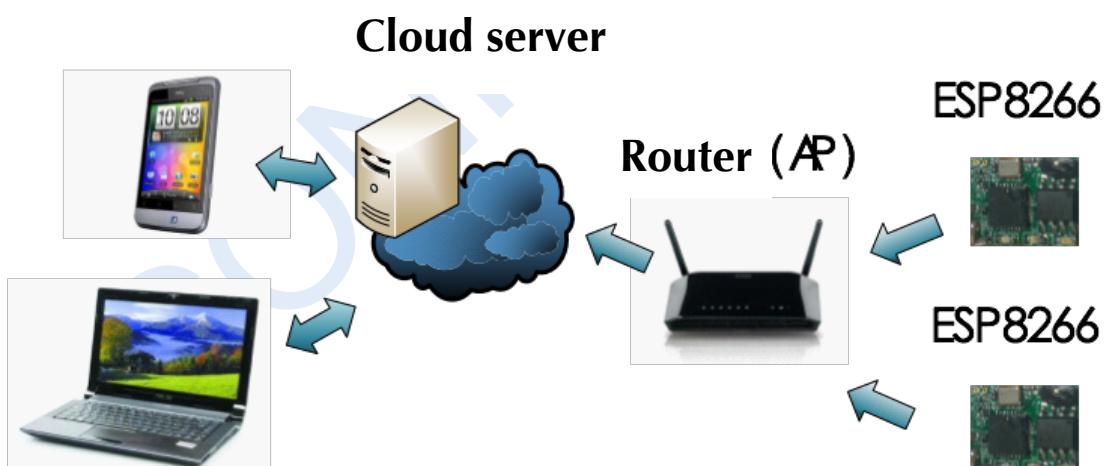
3.3.3. Transparent Serial

Example: Device's MCU as host, connects and transmits data to ESP8266EX through UART. ESP8266EX then transmits via W-Fi destination address, working in pass-through mode.

4. AT Instruction Set

4.1. Overview

Each instruction set contains four types of AT commands.

Test	A T+<x>=?	Query the Set command or internal parameters and its range values.
Query	A T+<x>?	Returns the current value of the parameter.
Set	A T+<x>=<...>	Set the value of user-defined parameters in commands and run.
Execute	A T+<x>	Runs commands with no user-defined parameters.

Note:
1. Not all AT instruction has four commands.
2. [] = default value, not required or may not appear
3. String values require double quotation marks, example AT +CWSAP="ESP756190","21030826",1,4
4. Baud rate = 115200

4.2. Basic AT Instructions

Command	Description	Response	Reference
AT	Test AT startup	OK	
AT+RST	Restart module	OK	
AT+GMR	View version info	<number> OK	<number> = 8-digit version no.

Espressif
Systems

4.3.   Wi-Fi Instructions

Command		Description	Response	Reference
AT+CWMODE		**Select Wi-Fi application mode**		<mode> 1: Station 2:AP 3. Both
	Test	AT+CWMODE=?	+CWMODE:(<mode> value list) OK	Set command requires reboot (AT+RST)
	Query	AT+CWMODE?	Back to current mode +CWMODE:<mode> OK	
	Set	AT+CWMODE=<mode>	OK	
AT+CWJAP		**Join Ap**		<ssid>=string value, AP name <pwd>=string value, max 64 ASCII chars
	Query	AT+CWJAP?	Back to selected AP +CWJAP:<ssid> OK	
	Set	AT +CWJAP=<ssid>,<pwd>	OK ERROR	
AT+CWLAP		**Lists available AP**		<ecn> 0: Open 1: WEP 2: WPA_PSK 3: WPA2_PSK 4: WPA_WPA2_PSK <ssid>=string value, AP name <rssi>=signal strength
	Execute	AT+CWLAP	Back to list of AP +CWLAP:<ecn>,<ssid>,<rssi> OK ERROR	

Espressif
Systems

Command	Description	Response	Reference
AT+CWQAP	**Disconnect from AP**		
Test	AT+CWQAP=?	OK	
Execute	AT+CWQAP	OK	
AT+ CWSAP	**Set parameters under AP mode**		Set command requires reboot
Query	AT+CWSAP?	Back to current AP parameters +CWSAP=<ssid>,<pwd>, <chl>,<ecn>	<ssid>=string value, AP name <pwd>=string value, max 64 ASCII chars <chl>=channel no. <ecn> 0: Open 1: WEP 2: WPA_PSK 3: WPA2_PSK 4: WPA_WPA2_PSK
Set	AT +CWSAP=<ssid>,<pwd>, <chl>,<ecn>	OK ERROR	
AT+ CWLIF	**Check IP of connected device**		
Execute	AT+CWLIF	<ip addr> OK	<ip addr>=IP address of connected device

September 17, 2014

4.4. TCP/IP Toolbox

Command	Description	Response	Reference
AT+CIPSTATUS	**Get connection status**		<id>=connected ID no. 0-4 <type>=string value, conection type "TCP" or "UDP" <addr>=string value, IP address <port>= port no. <tetype> 0: Connect as client 1: Connect as server
Execute	AT+CIPSTATUS	Back to current connection mode and status +CIPSTATUS:<id>,<type>, <addr>,<port>,<tetype> OK	
AT+CIPSTART	**Establish TCP connection or register UDP ports**		<id>=connected ID no. 0-4 <type>=string value, conection type "TCP" or "UDP" <addr>=string value, IP address of remote server <port>= port no. of remote server
Test	AT+CIPSTART=?	1. AT+CIPMUX=0 +CIPSTART:(<type>value list),(<IP address>range),(<port>range) +CIPSTART:(<type>value list),(<domain name>range),(<port>range) OK 2. AT+CIPMUX=1 +CIPSTART:(id),(<type>value list),(<IP address>range),(<port>range) +CIPSTART:(id),(<type>value list),(<domain name>range),(<port>range) OK	

Command	Description	Response	Reference
Set	1. Single Connection (+CIPMUX=0) AT +CIPSTART=<type>,<addr>,<port> 2. Multiple Connections (+CIPMUX=0) AT +CIPSTART=<id>,<type>, <addr>,<port>	Syntax correct and connection successful, returns OK else returns ERROR If connection already exists, returns ALREADY CONNECT	
AT+CIPSEND	**Send Data**		<id>= required ID no. of transmit connection <length>=numeric value, size of transmit packet, maximum 2048 bytes
Test	AT+CIPSTART=?	OK	
Set	1. Single Connection (+CIPMUX=0) AT+CIPSEND=<length> 2. Multiple Connections (+CIPMUX=0) AT +CIPSEND=<id>,<length>	Wrap return ">" after set command. Begins receive of serial data, when data length is met, starts transmission of data. If connection cannot be established or gets disconnected during send, returns ERROR If data is transmitted successfully, returns SEND OK	

Command	Description	Response	Reference
Execute	AT+CIPSEND	Wrap return ">" after execute command. Enters pass-through mode, 20ms interval between each packet, maximum 2048 bytes per packet. When single packet containing "+++" is received, it returns to command mode. Command can only be used in pass-through mode and single connection mode.	
AT+CIPCLOSE	**Close TCP or UDP connection**		\<id\>= required ID no. of connection to close, when id=5, all connections will be closed. (id=5 has no effect in server mode)
Test	AT+CIPCLOSE=?	OK	
Set	Multiply connection mode AT+CIPCLOSE=\<id\>	No errors, returns OK If connection \<id\> is disconnected, returns LINK IS NOT	
Execute	Single connection mode AT+CIPCLOSE	No errors, returns OK If no such connection, returns ERROR Prints UNLINK when there is no connection	

Espressif
Systems

| Command	Description	Response	Reference
AT+CIPSR	**Get local IP address**		\<IP address\>: device's current IP address (Station mode)
Test	AT+CIPSR=?	OK	No effect under AP mode
Execute	AT+CIPSR	+CIFSR:\<IP address\> OK ERROR	
AT+CIPMUX	**Start multiple connections**		\<mode\> 0: Single Connection 1: Multiple Connection
Query	AT+CIPMUX?	+CIPMUX:\<mode\> OK	Mode can only be changed after all connections are disconnected. If server is started, reboot is required.
Set	AT+CIPMUX=\<mode\>	OK If already connected, returns LINK IS BUILDED	

September 17, 2014

Espressif
Systems

Command	Description	Response	Reference
AT+ CIPSERVER	**Configure as server**		
Set	AT +CIPSERVER=<mode>[,< port>]	OK Server has to be restarted	<mode> 0: Server mode OFF 1: Server mode ON <port>: port number, default value=333 Server monitor will automatically be created when Server mode ON. When a client is connected to the server, it will take up one connection. Server can only be turned ON when AT +CIPMUX=1
AT+ CIPMODE	**Set module transfer mode**		<mode> 0: No-passthrough 2: Passthrough
Query	AT+CIPMODE?	+CIPMODE:<mode> OK	
Set	AT+CIPMODE=<mode>	OK If already connected, returns LINK IS BUILDED	

Command		Description	Response	Reference
AT+CIPSTO		**Set server timeout**		<time>: Server timeout Range 0~28800 Units in seconds
	Query	AT+CIPSTO?	+CIPSTO:<time> OK	
	Set	AT+CIPSTO=<time>	OK	
+IPD		**Receive network data**		
		1. Single Connection (+CIPMUX=1) +IPD,<len>:<data> 2. Multiple Connection (+CIPMUX=1) +IPD,<id>,<len>,<data>	When the module receives network data, it will send the data through the serial port using +IPD command	<id>: id no. given by connection <len>: length of data <data>: received data Only valid in command mode

September 17, 2014
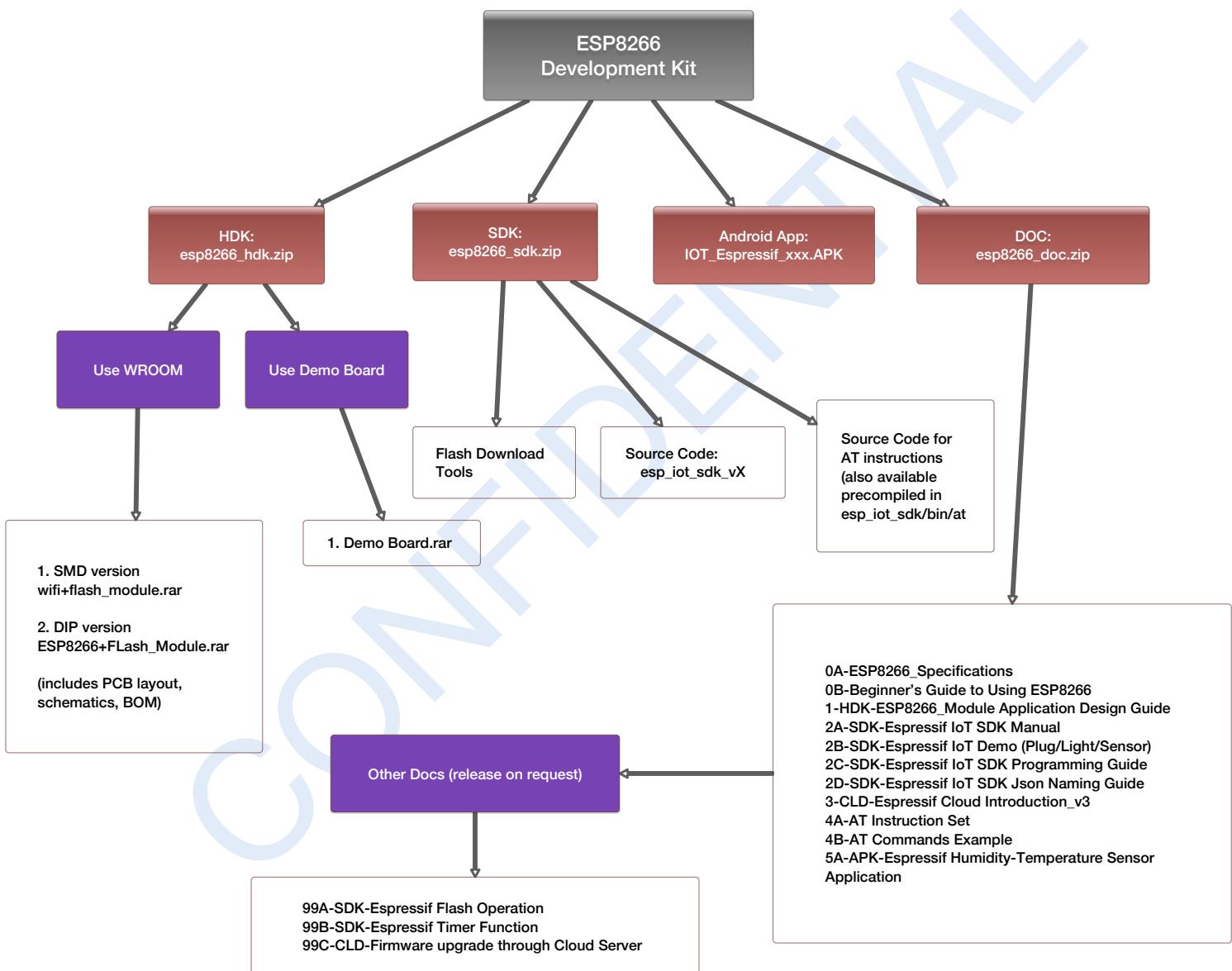
5.2. Documentation List

Espressif provides reference guides to help developers use ESP8266EX in their projects:

- **ESP8266 Module Application Design Guide**: How to develop ESP8266EX modules using our HDK
- **Espressif IoT SDK Manual**: ESP8266EX development environment
- **Espressif IoT Demo**: ESP8266EX applications in Plug/Light/Sensor
- **Espressif IoT SDK Programming Guide**: Using the SDK
- **Espressif Cloud Introduction**: How to use Espressif Cloud Server
- **Espressif Humidity-Temperature Sensor Application[1]**: User guide to our in-house developed Andriod application.
- **Espressif Flash Operation**: Technical doc, ESP8266EX Flash RW
- **Espressif Timer Function**: Technical doc, ESP8266EX timer applications
- **Firmware upgrade through Cloud Server:** Technical doc, How to upgrade firmware of ESP8266EX using Cloud.

---

[1] APK and documentation is not available in English yet.

Appendix: Contact Details

For more information, you can contact us at

Address:

456 BI BO ROAD SUITE A201

PUDONG, SHANGHAI

CHINA 201203

Email:

SALES@ESPRESSIF.COM

SUPPORT@ESPRESSIF.COM

Corporate Website:

HTTP://WWW.ESPRESSIF.COM