ESPRESSIF
ESP8266EX:
A BEGINNER'S GUIDE

[√] CONFIDENTIAL

[ ] INTERNAL

[ ] PUBLIC

Table of Contents

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.

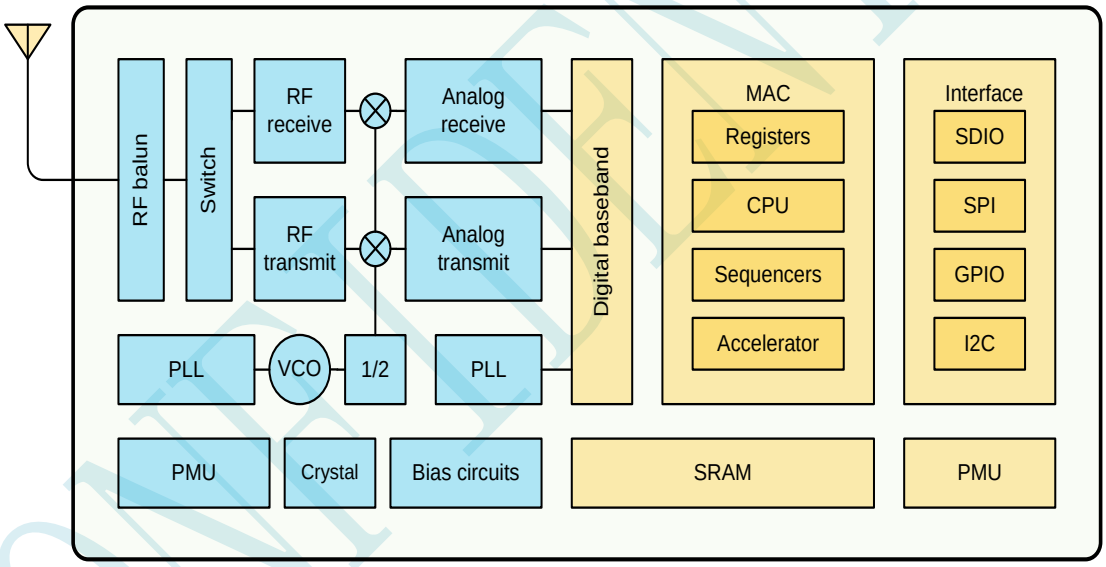


Figure 1: ESP8266EX Block Diagram

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 & 0.4 s 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	802.11 b/g/n
	Frequency	2.4G-2.5G(2400M-2483.5M)
	Tx Power	802.11 b: 20 dBm
		802.11 g: 17 dBm
		802.11 n: 14 dBm
	Rx Sensitivity	802.11 b: (11Mbps) -91dbm
		802.11 g: (54Mbps) -75dbm
		802.11 n: (MCS7) -72dbm
	Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip
Hardware	Peripheral Bus	UART/SDIO/SPI/I2C/

		GPIO/PWM
	Operating Voltage	3.0~3.6V
	Operating Current	Avg. 80mA
	Operating Temperature	-40° ~125°
	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 Download /OTA(via network)
	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

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

1.2. Hardware Overview

1.2.1. ESP8266EX Pin Definition

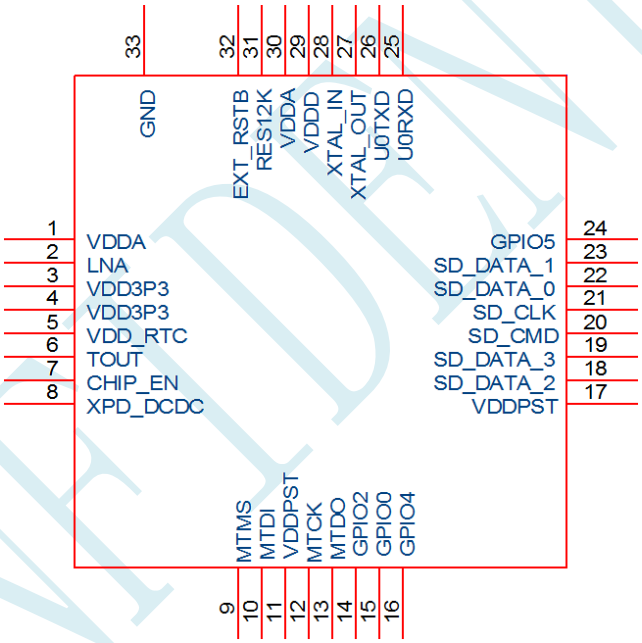


Figure 2: ESP8266EX Pin Definition Diagram

Table 1 ESP8266EX Pin Function

Pin	Name	Type	Function
1	VDDA	P	Analog Power 3.0 ~3.6V
2	LNA	I/O	RF Antenna Interface, Chip Output Impedance=50Ω 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 programming; 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 programming; 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

1.2.2. Electrical Characteristics

Table 2 ESP8266 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
I/O	V_{IL}/V_{IH}		-0.3/0.75V_{IO}		0.25V_{IO}/3.6	V
	V_{OL}/V_{OH}		N/0.8V_{IO}		0.1V_{IO}/N	
	I_{MAX}				12	mA
Electrostatic Discharge (HBM)		TAMB=25℃			2	KV
Electrostatic Discharge (CDM)		TAMB=25℃			0.5	KV

Table 3 ESP8266 Power Consumption

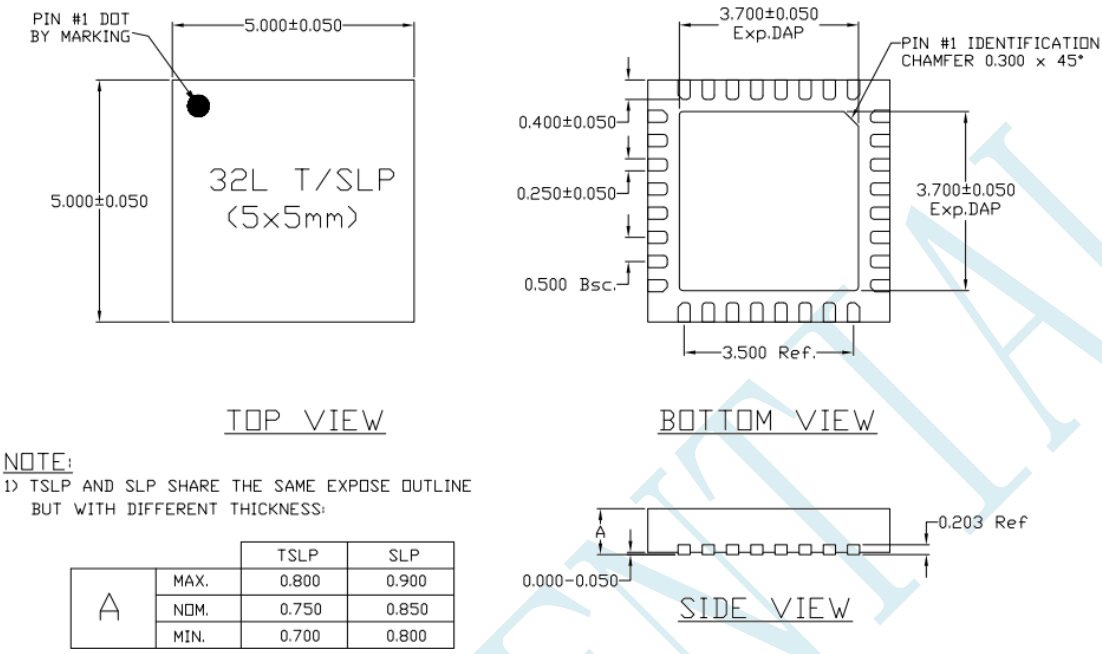
Parameter	Typical	Unit
Tx802.11b, CCK 11Mbps, P OUT=+17dBm	170	mA
Tx 802.11g, OFDM 54Mbps, P OUT =+15dBm	140	mA
Tx 802.11n, MCS7, P OUT =+13dBm	120	mA
Rx 802.11b, 1024 bytes packet length , -80dBm	50	mA
Rx 802.11g, 1024 bytes packet length, -70dBm	56	mA
Rx 802.11n, 1024 bytes packet length, -65dBm	56	mA
Modem-Sleep①	15	mA
Light-Sleep②	0.9	mA
Deep-Sleep③	10	uA
Power 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.

1.2.3. ESP8266 QFN32 Package Footprint



		TSLP	SLP
A	MAX.	0.800	0.900
	NOM.	0.750	0.850
	MIN.	0.700	0.800

Figure 3： ESP8266EX QFN32 Package Dimensions

1.2.4. Hardware Development Kit

Espressif provides a demo board for ESP8266EX.

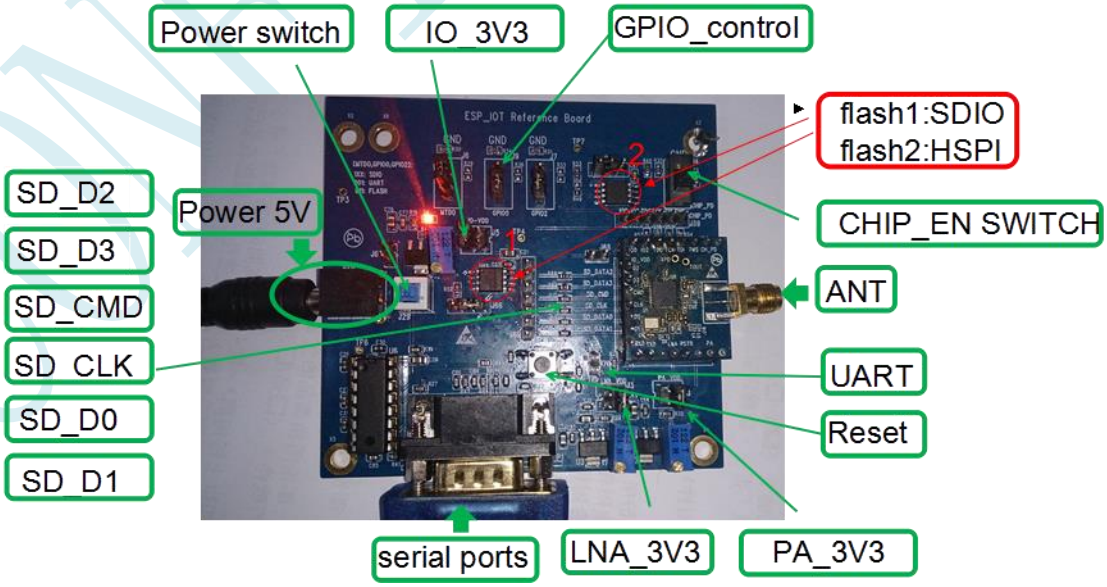


Figure 4: ESP8266EX Demo Board (Brief Outline)

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

2、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.

3、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.

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

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

6、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.

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

7、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.

1.2.5. ESP8266EX Modules(WROOM)

Espressif offers 3 kinds of modules: SMD ,DIP and module for development .

(1) SMD:

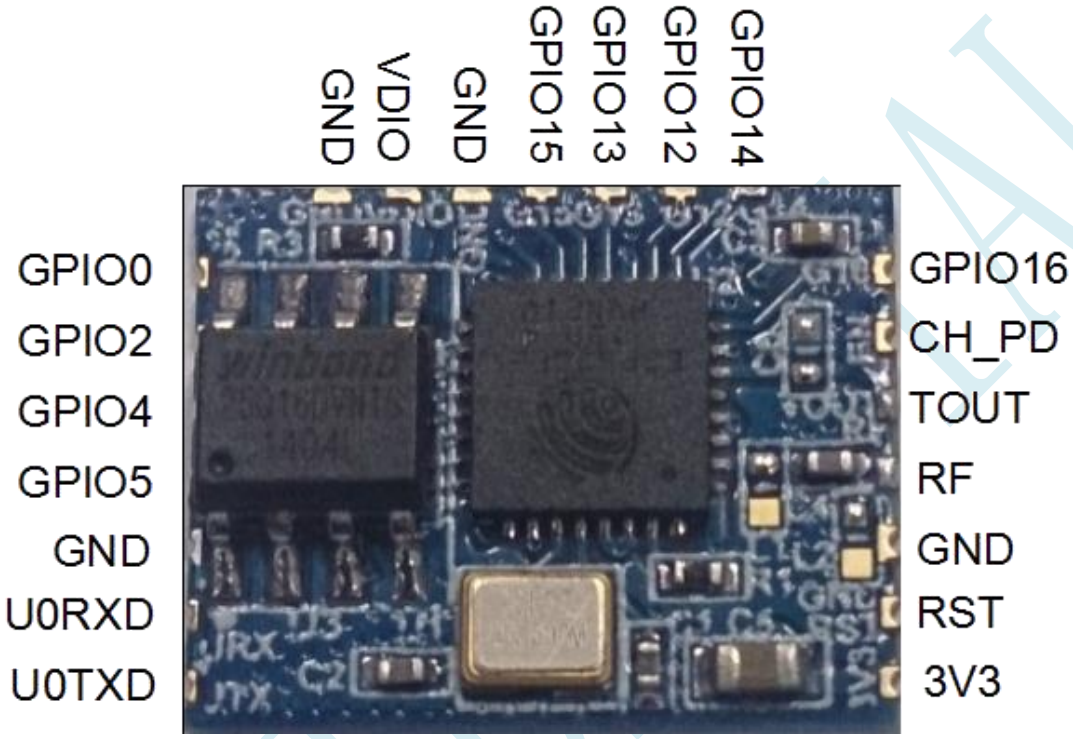


Figure 5: WROOM SMD Module

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):

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

2、CH_PD: high.

3、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.

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

(2)DIP:

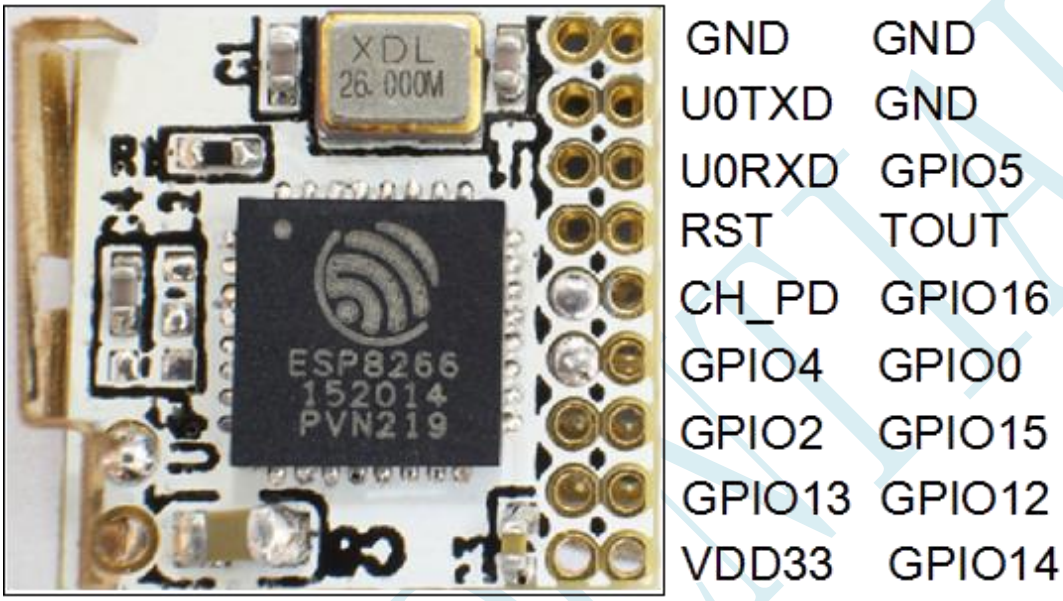


Figure 6: WROOM DIP Module

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.

(3)Module for development：

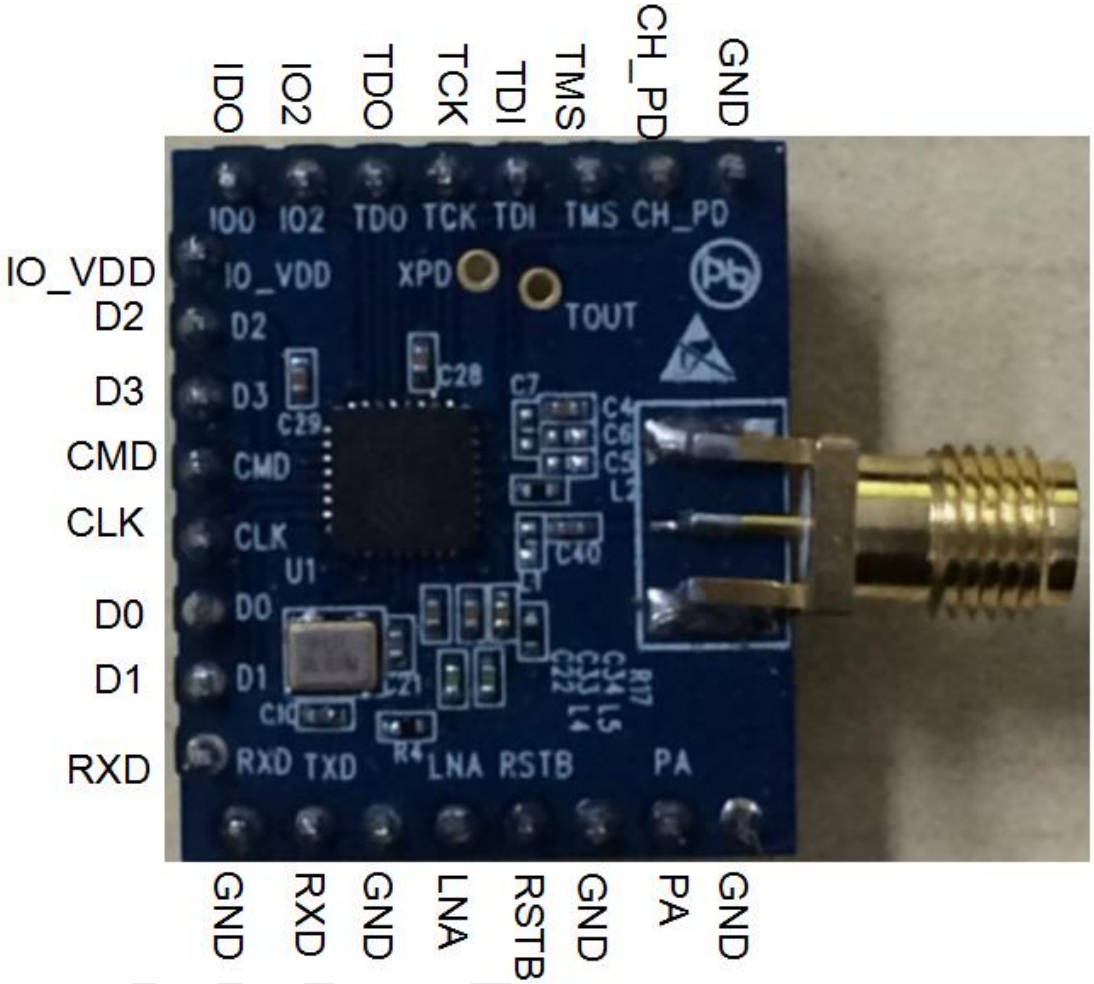


Figure 7： Module for Development

Module size is 22x25.5mm. Refer below for usage of module (Tables in section 1.2.1 for reference):

1、Connect IO_VDD、LNA、PA to an external power source.

2、CH_PD : high.

3、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.

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

1.3. Applications using ESP8266EX

1.3.1.2 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 8 shows a demo board of temperature-humidity sensor. All sensors connect to ESP8266EX by I2C.

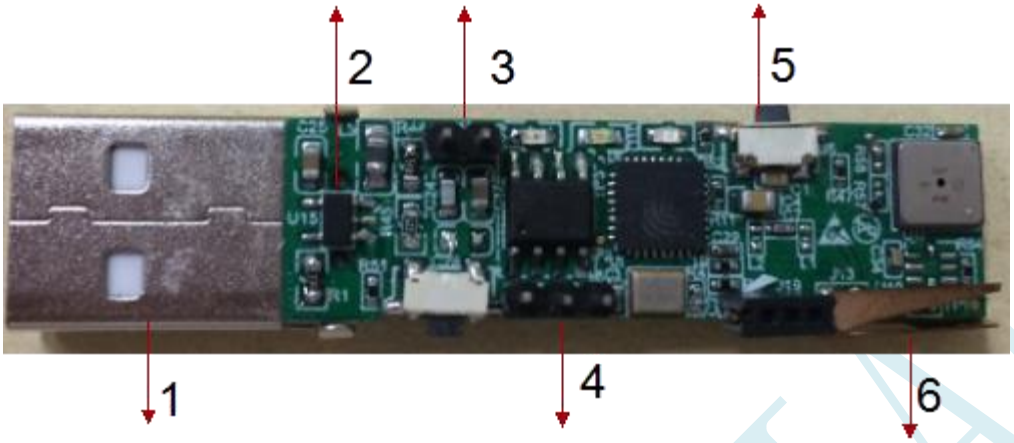


Figure 8: ESP8266EX as used in USB Sensor Demo

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

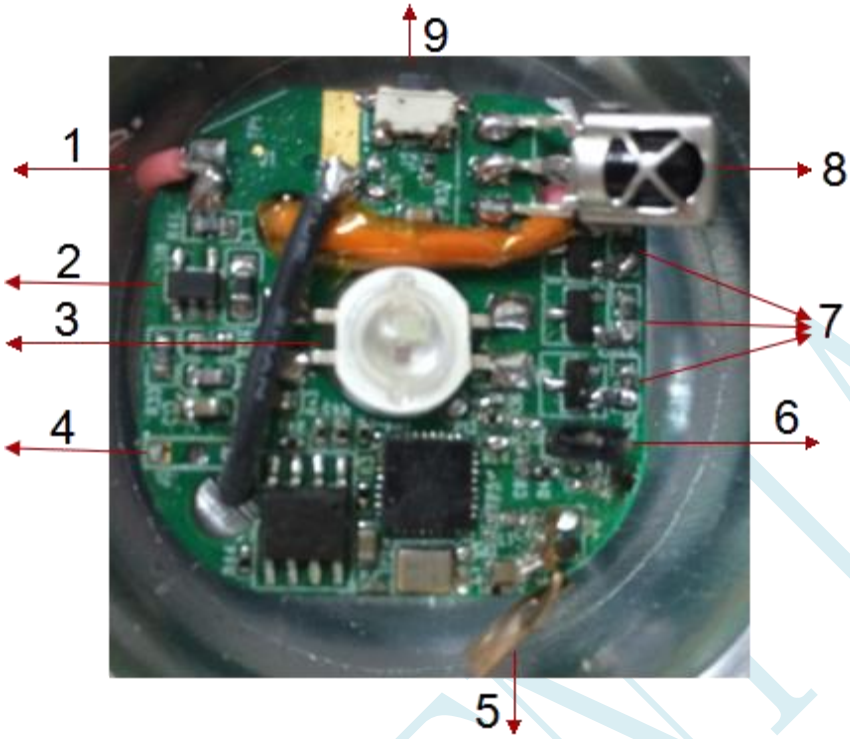


Figure 9: ESP8266EX as used in Smart Light Demo

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

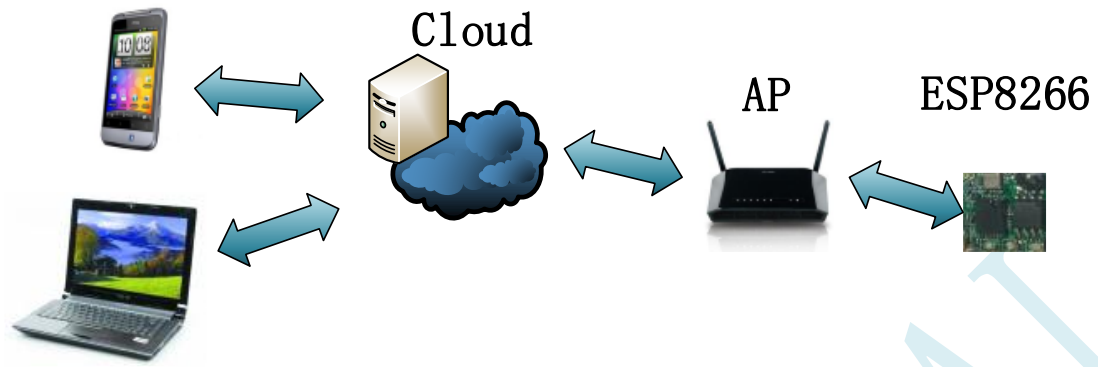
Use ESP8266EX to achieve a flexible network topology.

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.

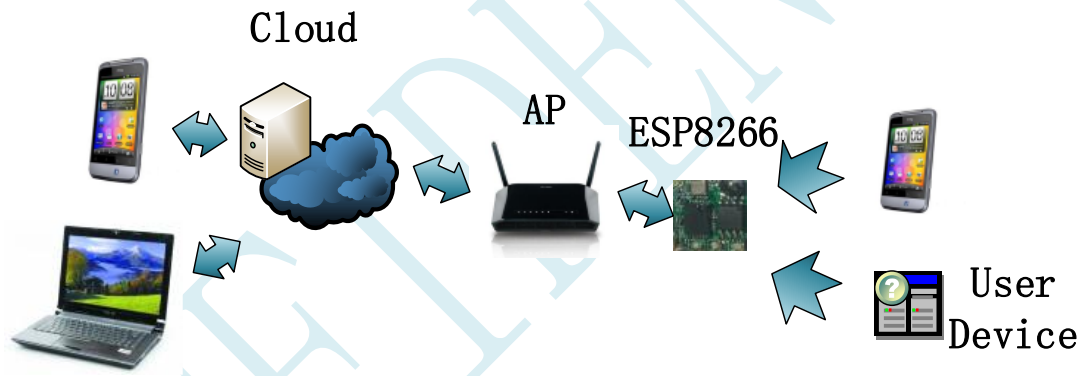
2.1.2. Station Mode

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 Transmission

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 transmission	
Wireless network parameters (AP INFO)	SSID
	Encryption
	Password
TCP connection parameters	Protocol type
	Connection type (client)
	Destination IP address and 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

ESP8266 EX supports many different kinds of encryption:

- WEP (only in station mode)
- WPA-PSK/TKIP
- WPA-PSK/AES
- WPA2-PSK/TKIP
- WPA2-PSK/AES

2.5. Low Power Operation

ESP8266EX supports 3 low power operation modes:

Mode	Modem-Sleep	Light-Sleep	Deep-Sleep
Action	Turn off WiFi modem. CPU and other peripherals are still running.	Turn off WiFi 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	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	Used in applications where Wi-Fi need not be always connect, a long time before transmitting the application data packets. Measured once

	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.	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. "	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 Update

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

1） Goto http://iot.espressif.cn/#/

"Start"->"Developer API": API device control instructions.

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

"Register"： New user registration.

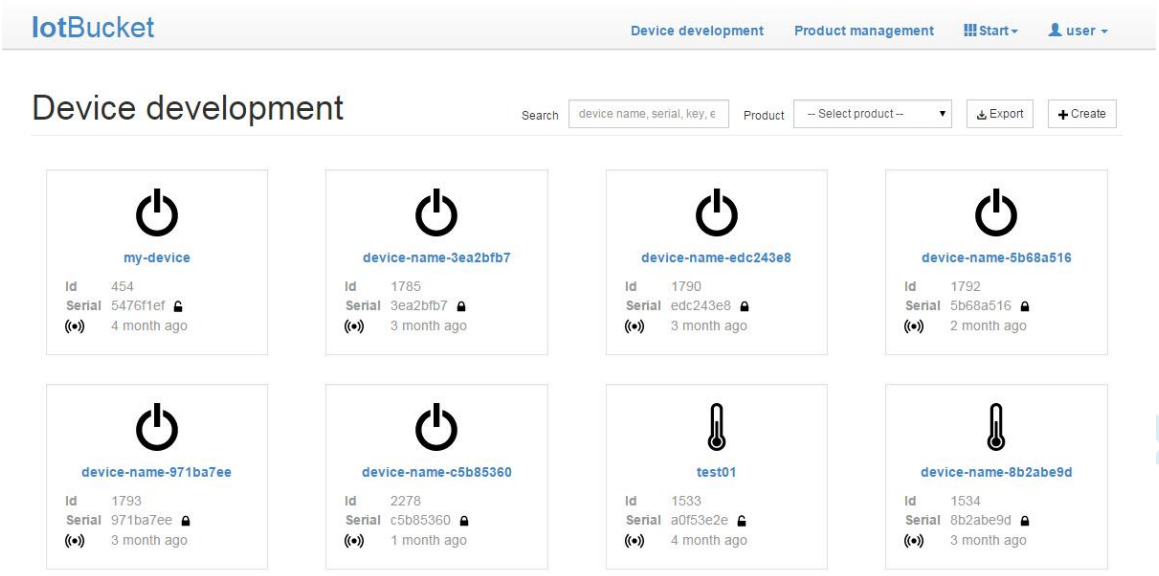
"Login"： Login if registered user.

2） After logging in, click on "Device Development". User can edit the settings for all the devices.

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

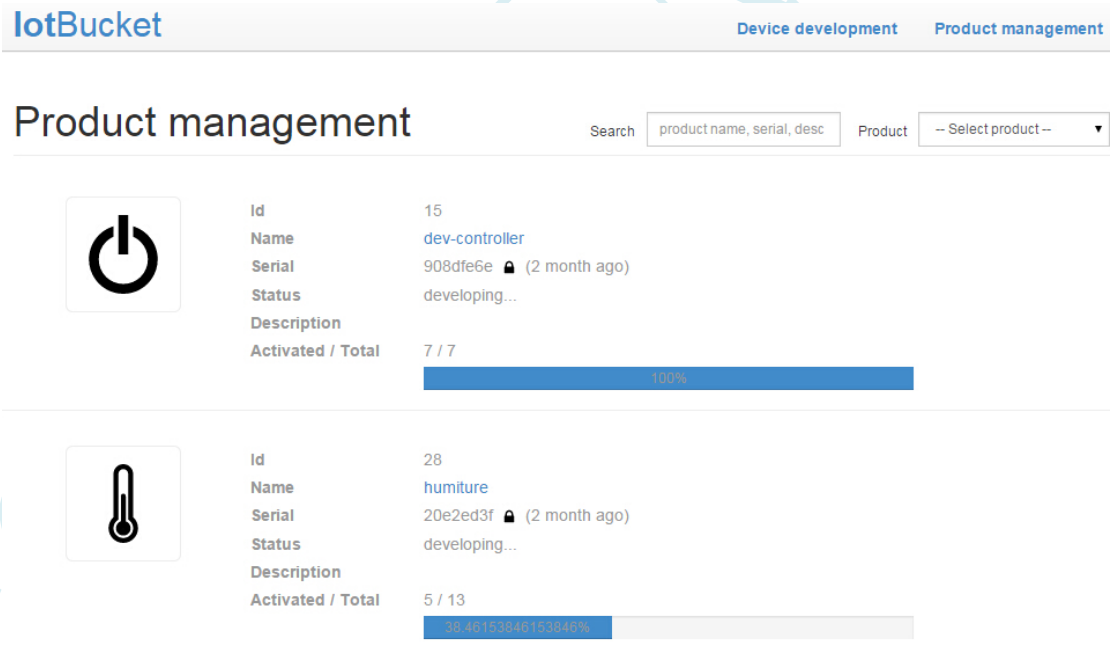
"Export" : Export the device list

"Create" : Create a new device



3） "Product Management" shows the list of products.

Use "Search", "Product" to filter and query products.



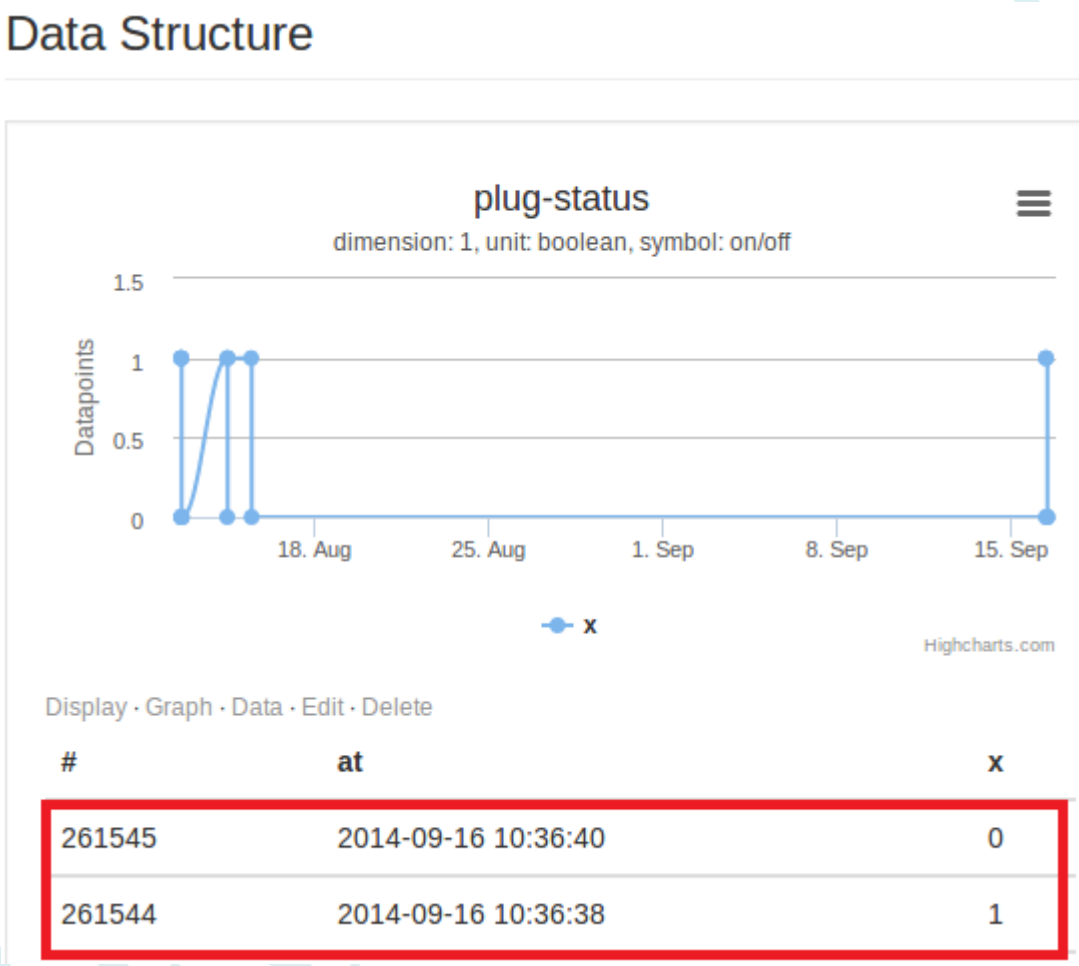
3.1.1. Device Development

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 Structure" -> "Data" to view device history.



3.1.1.2. Debugging info

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

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

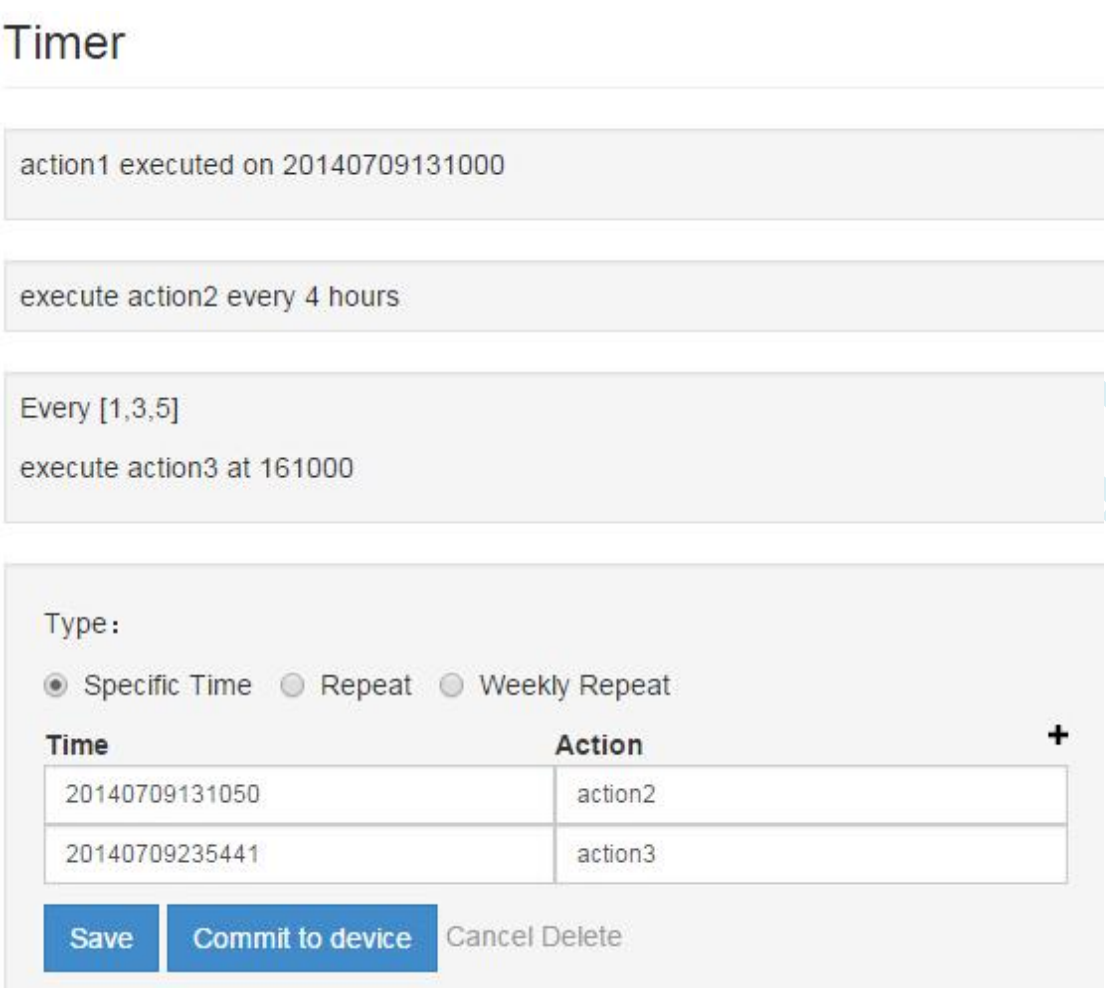
3.1.1.3. Timer

Espressif Cloud server supports 3 kinds of timers:

1） Specific timer: Execute a command at a specified time

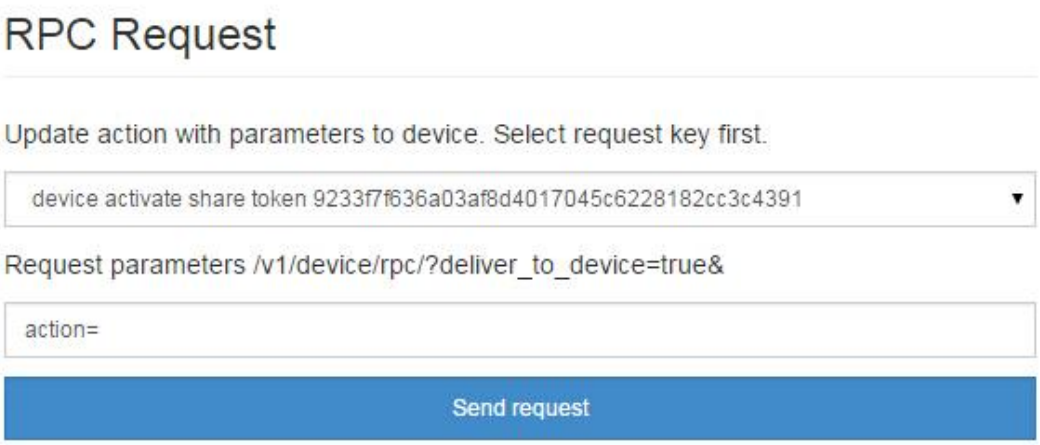
2） Repeat Timer: Execute a command at fixed intervals

3） Weekly Repeat Timer: Execute a command at weekly or on certain days each week



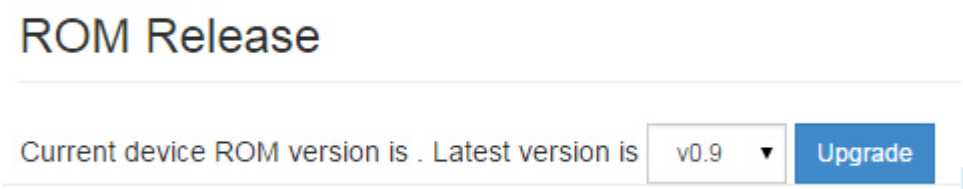
3.1.1.4. Customized Reverse Control

User can customize an action to reverse control the device.



3.1.1.5. Firmware upgrade (OTA)

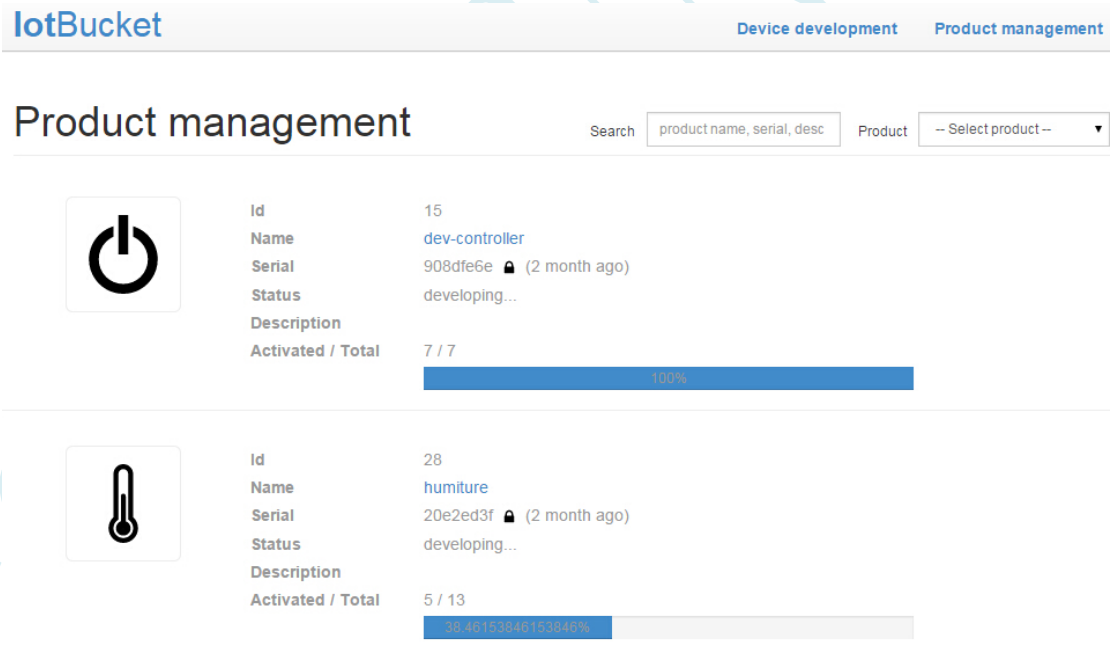
User update the firmware through Cloud Server



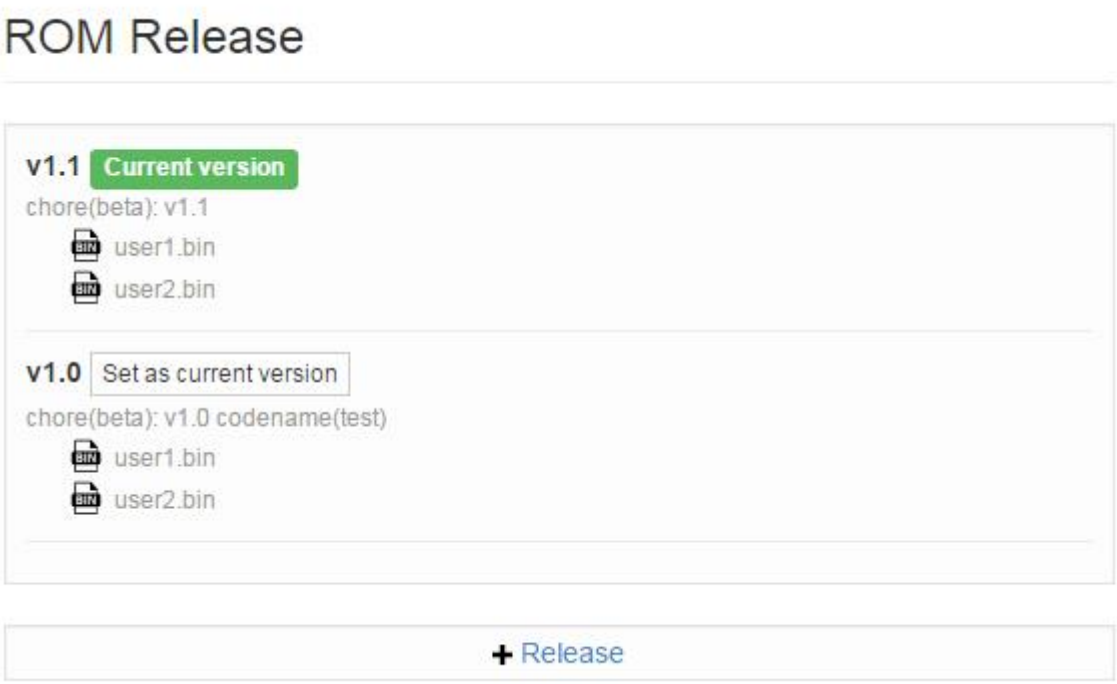
3.1.2. Product management

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

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



2） 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 Release
v1.1
Current version
chore(beta): v1.1
user1.bin
user2.bin
v1.0
Set as current version
chore(beta): v1.0 codename(test)
user1.bin
user2.bin
+ Release

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)

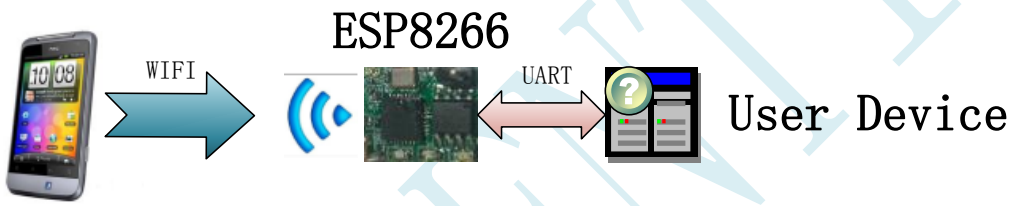
- Default Encryption: WPA/WPA2
- Default Serial Parameters: 74880, 8, 1, None

- Default IP Address: 192.168.4.1

3.3. Application Examples

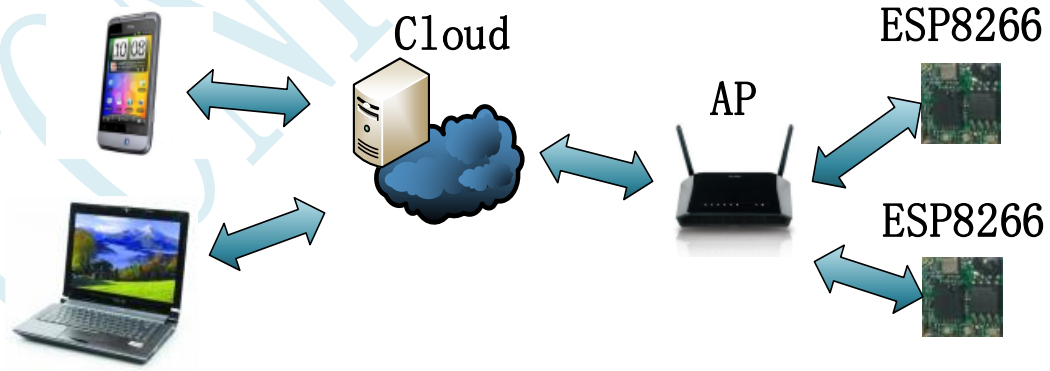
3.3.1. Wi-Fi Remote Control in LAN

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 through Cloud

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 Transmission

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

4. AT Instruction Set

This part introduces Espressif AT command instruction set and usage.

4.1. Overview

Instruction set is divided into: Basic AT commands, WiFi function, AT commands, TCP / IP Toolbox AT commands.

4.1.1. Instruction Description

Each instruction set contains four types of AT commands.

Type	Instruction Format	Description
Test	AT+<x>=?	Query the Set command or internal parameters and its range values.
Query	AT+<x>?	Returns the current value of the parameter.
Set	AT+<x>=<…>	Set the value of user-defined parameters in commands and run.
Execute	AT+<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, for example: AT+CWSAP="ESP756190","21030826",1,4
4. Baud rate = 115200
5. AT instruction ends with "\r\n"

4.1.2. AT Instruction Listing

Instructions	Description
Basic	
AT	Test if AT startup
AT+RST	Restart
AT+GMR	View version info
Wi-Fi	
AT+CWMODE	WIFI mode（station/softAP/station+softAP）
AT+CWJAP	Connect to AP
AT+CWLAP	Lists available APs
AT+CWQAP	Disconnect from AP
AT+CWSAP	Set parameters under AP mode
AT+CWLIF	Get stations' ip which are connected to ESP8266 softAP
TCP/IP	
AT+CIPSTATUS	Get connection status
AT+CIPSTART	Establish TCP connection or register UDP port
AT+CIPSEND	Send data
AT+CIPCLOSE	Close TCP/UDP connection
AT+CIFSR	Get local IP address
AT+CIPMUX	Set multiple connections mode
AT+CIPSERVER	Configure as server
AT+CIPMODE	Set transmission mode
AT+CIPSTO	Set timeout when ESP8266 runs as TCP server
AT+CIUPDATE	For OTA（upgrade through network）
Data RX	
+IPD	Data received from network

4.2. Basic AT Instruction Set

List of basic AT instructions:

Basic	
Instruction	**Description**
AT	Test AT startup
AT+RST	Restart module
AT+GMR	View version info

4.2.1. AT – Test AT startup

AT – Test AT startup	
Type： execute Instruction： **AT**	Response： OK
	Parameter description： null

4.2.2. AT+RST – Restart module

AT+RST – Restart module	
Type ： execute Instruction： **AT+RST**	Response： OK
	Parameter description： null

4.2.3. AT+GMR – View version info

AT+GMR – View version info	
Type ： execute Instruction：	Response： <number>

AT+GMR	OK
	Parameter description： < number > version info，length：8 bytes
Note	For example，response is 0017xxxxxx，then 0017 means the AT version.

4.3. WIFI functions

WIFI	
Instruction	**Description**
AT+CWMODE	WIFI mode（station/softAP/station+softAP）
AT+CWJAP	Connect to AP
AT+CWLAP	Lists available APs
AT+CWQAP	Disconnect from AP
AT+ CWSAP	Set parameters under AP mode
AT+ CWLIF	Get station's ip which is connected to ESP8266 softAP

4.3.1. AT+CWMODE – WIFI mode

AT+CWMODE - WIFI mode（station/softAP/station+softAP）	
Type：test Function: Get value scope of wifi mode. Instruction： **AT+CWMODE=?**	Response： +CWMODE:(value scope of <mode>) OK
	Parameter description： <mode>1 means Station mode 2 means AP mode 3 means AP + Station mode
Type：query	Response：

Function: Query ESP8266's current wifi mode. Instruction: **AT+CWMODE?**	+CWMODE:<mode> OK Parameter description: The same as above.
Type: set Function: Set ESP8266 wifi mode Instruction: **AT+CWMODE=<mode>**	Response: OK Parameter description: The same as above.

4.3.2. AT+CWJAP – Connect to AP

AT+CWJAP – Connect to AP	
Type: query Function: Query AP's info which is connect by ESP8266. Instruction: **AT+ CWJAP?**	Response: + CWJAP:<ssid> OK Parameter description: <ssid> string, AP's SSID
Type: set Function: Set AP's info which will be connect by ESP8266. Instruction: **AT+ CWJAP =<ssid>,< pwd >**	Response: OK ERROR Parameter description: <ssid> string, AP's SSID

	<pwd> string, MAX: 64 bytes

4.3.3. AT+CWLAP – List available APs

AT+CWLAP - Lists available APs	
Type：set Function: Search available APs with specific conditions. Instruction： **AT+ CWLAP =** **<ssid>,< mac >,<ch>**	Response： + CWLAP: <ecn>,<ssid>,<rssi>,<mac> OK ERROR Parameter description： The same as below.
Type ： execute Function: Lists all available APs. Instruction： **AT+CWLAP**	Response： + CWLAP: <ecn>,<ssid>,<rssi>,<mac> OK ERROR Parameter description： < ecn >0 OPEN 1 WEP 2 WPA_PSK 3 WPA2_PSK 4 WPA_WPA2_PSK <ssid> string, SSID of AP <rssi> signal strength <mac> string, MAC address

4.3.4. AT+CWQAP – Disconnect from AP

AT+CWQAP - Disconnect from AP	
Type：test Function: Only for test Instruction： **AT+CWQAP=?**	Response： OK Parameter description：
Type ： execute Function: Disconnect from AP. Instruction： **AT+ CWQAP**	Response： OK Parameter description：

4.3.5. AT+ CWSAP – Configuration of softAP mode

AT+ CWSAP – Configuration of softAP mode	
Type：Query Function: Query configuration of softAP mode. Instruction： **AT+ CWSAP?**	Response： + CWSAP:<ssid>,<pwd>,<chl>,<ecn> Parameter description： The same as below.
Type：Set Function: Set configuration of softAP mode.	Response： OK ERROR

Instruction： **AT+ CWSAP=** **<ssid>,<pwd>,<chl>,<ecn>**	Note: This CMD is only available when softAP mode enable, and need to follow by AT+RST to make it works. Parameter description： <ssid> string, ESP8266 softAP' SSID <pwd> string, MAX: 64 bytes <chl> channel id < ecn >0 OPEN 2 WPA_PSK 3 WPA2_PSK 4 WPA_WPA2_PSK

4.3.6. AT+ CWLIF – IP of stations

AT+ CWLIF – ip of stations which are connected to ESP8266 softAP	
Type ： execute Function: Get ip of stations which are connected to ESP8266 softAP Instruction： **AT+CWLIF**	Response： <ip addr> OK Parameter description： <ip addr> ip address of stations which are connected to ESP8266 softAP

4.4. TCP/IP Related

TCP/IP	
Instruction	**Description**
AT+ CIPSTATUS	Get connection status
AT+CIPSTART	Establish TCP connection or register UDP port
AT+CIPSEND	Send data
AT+CIPCLOSE	Close TCP/UDP connection
AT+CIFSR	Get local IP address
AT+CIPMUX	Set multiple connections mode
AT+CIPSERVER	Configure as server
AT+CIPMODE	Set transmission mode
AT+CIPSTO	Set timeout when ESP8266 runs as TCP server

4.4.1. AT+ CIPSTATUS – Information about connection

AT+ CIPSTATUS – Information about connection	
Type ： execute Function: Get information about connection. Instruction: **AT+ CIPSTATUS**	Response： STATUS:<stat> + CIPSTATUS:<id>,<type>,<addr>,<port>,<tetype> OK
	Parameter description： <stat> 2: Got IP 3: Connected 4: Disconnected

	<id> id of the connection (0~4), for multi-connect <type> string, "TCP" or "UDP" <addr> string, IP address. <port> port number <tetype> 0: ESP8266 runs as client 1: ESP8266 runs as server

4.4.2. AT+CIPSTART – Start connection

AT+CIPSTART – Establish TCP connection or register UDP port, start connection	
Type ： test Function: Get the information of parameter. Instruction: **AT+CIPSTART=?**	Response： 1) If AT+CIPMUX=0 +CIPSTART:(<type>),(<IP address>),(<port>) +CIPSTART:(<type>),(<domain name>),(<port>) OK 2) If AT+CIPMUX=1 +CIPSTART:(id),(<type>),(<IP address>),(<port>) +CIPSTART: (id), (<type>),(<domain name>),(<port>)
	Parameter description： null
Type ： Set Function: Start a connection as client. Instruction:	Response： OK or ERROR If connection already exists, returns

1)Single connection **(+CIPMUX=0)** **AT+CIPSTART=** **<type>,<addr>,<port>** 2)Multiple connection (+CIPMUX=1) **AT+CIPSTART=** **<id><type>,<addr>,<port>**	ALREAY CONNECT Parameter description： <id> 0-4 , id of connection <type> string, "TCP" or "UDP" <addr> string, remote ip <port> string, remote port

4.4.3. AT+CIPSEND – Send data

AT+CIPSEND – Send data	
Type ： test Function: Only for test. Instruction: **AT+CIPSEND=?**	Response： OK Parameter description： null
Type ： Set Function: Set length of the data that will be sent. For normal send. Instruction:	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

1)For single connection: (+CIPMUX=0) **AT+CIPSEND=<length>** 2) For multiple connection: (+CIPMUX=1) **AT+CIPSEND=<id>,<length>**	disconnected during send, returns ERROR If data is transmitted successfully, returns SEND OK Note: This CMD Parameter description： <id> ID no. of transmit connection <length> data length, MAX 2048 bytes
Type ： execute Function: Send data. For unvarnished transmission mode. Instruction: **AT+CIPSEND**	Response： Wrap return ">" after execute command. Enters unvarnished transmission, 20ms interval between each packet, maximum 2048 bytes per packet. When single packet containing "+++" is received, it returns to command mode. This command can only be used in unvarnished transmission mode which require to be single connection mode.

4.4.4. AT+CIPCLOSE – Close TCP or UDP connection

AT+CIPCLOSE – Close TCP or UDP connection	
Type ： test Function: Only for test. Instruction:	Response： OK

AT+CIPCLOSE=?	
Type ： Set Function: Close TCP or UDP connection. Instruction:	Response： No errors, returns OK If connection <id> is disconnected, returns Link is not
For multiply connection mode **AT+CIPCLOSE=<id>**	Parameter description： <id> ID no. of connection to close, when id=5, all connections will be closed. (id=5 has no effect in server mode)
Type ： execute Instruction: For single connection mode **AT+CIPCLOSE**	Response： OK or If no such connection, returns ERROR Prints UNLINK when there is no connection

4.4.5. AT+CIFSR – Get local IP address

AT+CIFSR – Get local IP address	
Type ： Test Function: Only for test. Instruction:	Response： OK

AT+CIFSR=?	
Type ： Execute Function: Get local IP address. Instruction: **AT+ CIFSR**	Response： + CIFSR:<IP address> + CIFSR:<IP address> OK ERROR
	Parameter description： <IP address> IP address of ESP8266 softAP IP address of ESP8266 station

4.4.6. AT+ CIPMUX – Enable multiple connections

AT+ CIPMUX – Enable multiple connections or not	
Type ： Query Function: Get Parameter config. Instruction: **AT+ CIPMUX?**	Response： + CIPMUX:<mode> OK
	Parameter description： The same as below.
Type ： Set Function: Set connection mode. Instruction: **AT+ CIPMUX=<mode>**	Response： OK If already connected, returns Link is builded
	Parameter description：

	<mode>0 single connection 1 multiple connection
Note	This mode can only be changed after all connections are disconnected. If server is started, reboot is required.

4.4.7. AT+ CIPSERVER – Configure as server

AT+ CIPSERVER – Configure as server	
Type : Set Function: Set server. Instruction: **AT+ CIPSERVER=<mode>[,<port>]**	Response: OK Parameter description: <mode> 0 Delete server (need to follow by restart) 1 Create server <port> port number, default is 333
Note	1、Server can only be created when AT+CIPMUX=1 2、Server monitor will automatically be created when Server is created. 3、When a client is connected to the server, it will take up one connection, be gave an id.

4.4.8. AT+ CIPMODE – Set transfer mode

AT+ CIPMODE – Set transfer mode	
Type : Query	Response:

Function: Query transfer mode. Instruction: **AT+ CIPMODE?**	+ CIPMODE:<mode> OK Parameter description： The same as below.
Type ： Set Function: Set transfer mode. Instruction: **AT+CIPMODE=<mode>**	Response： OK If already connected, returns Link is builded Parameter description： <mode>0 normal mode 1 unvarnished transmission mode

4.4.9. AT+ CIPSTO – Set server timeout

AT+ CIPSTO – Set server timeout	
Type ： Query Function: Query server timeout. Instruction: **AT+CIPSTO?**	Response： + CIPSTO:<time> OK Parameter description： The same as below.
Type ： Set Function: Set server timeout. Instruction:	Response： OK Parameter description：

AT+CIPSTO=<time>	< time> server timeout, range 0~28800 seconds

4.4.10. AT+ CIUPDATE – Update through network

AT+ CIUPDATE – update through network	
Type ： execute Function: Start upgrade. Instruction: **AT+ CIUPDATE**	Response： +CIPUPDATE:<n> OK
	Parameter description： <n> 1 found server 2 connect server 3 got edition 4 start update

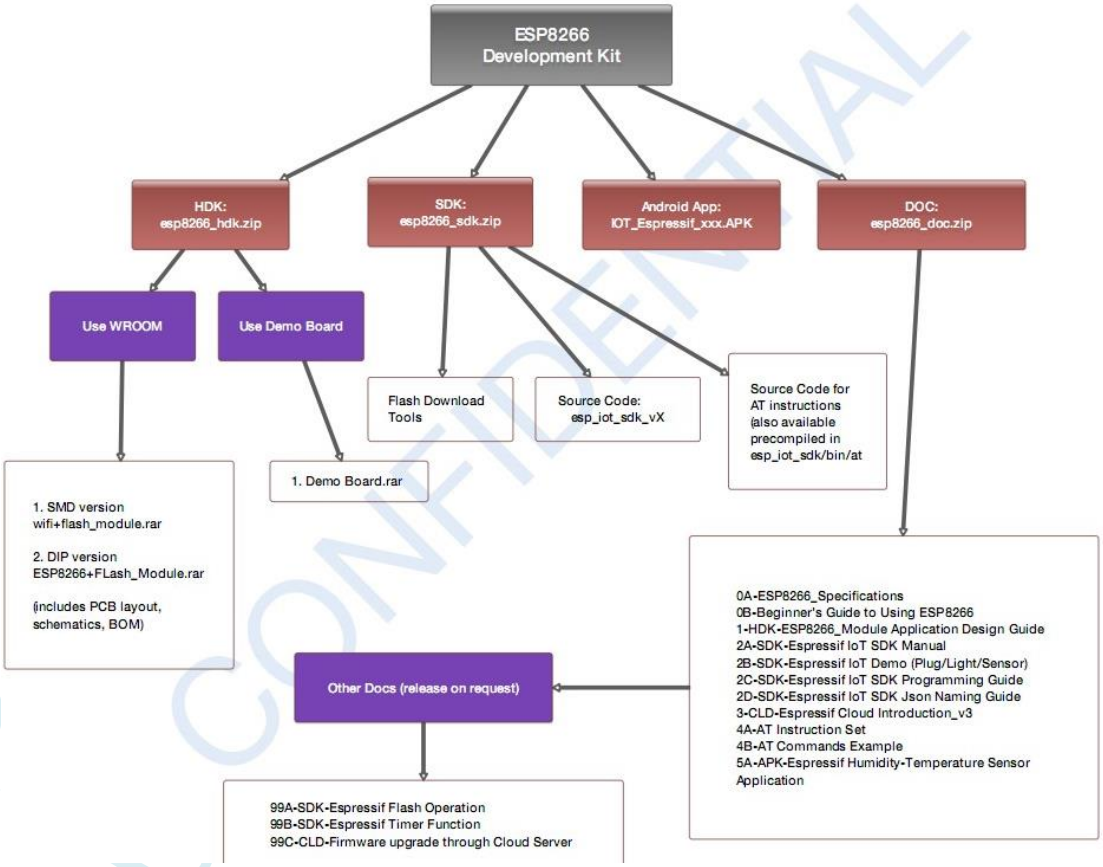
4.4.11. +IPD – Receive network data

+IPD – Receive network data	
1)Single connection: (+CIPMUX=0) **+IPD,<len>:<data>** 2) Multiple connection (+CIPMUX=1) **+IPD,<id>,<len>:<data>**	NOTE: When the module receives network data, it will send the data through the serial port using +IPD command Parameter description： <id> id no. of connection <len> data length <data> data received

5. Development Kit

5.1. Components of ESP8266EX Development

The ESP8266 development kit consists of a HDK (hardware development kit), SDK (software development kit), APK (android application) and DOC (user guides).



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 IoT phone APP Manual:** User guide to our in-house developed Android application.
- **Espressif IOT Flash RW Operation**: Technical doc, ESP8266EX Flash RW
- **Espressif Timer Function**: Technical doc, ESP8266EX timer applications
- **Firmware upgrade through Cloud (OTA)**: Technical doc, How to upgrade firmware of ESP8266EX using Cloud.

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

<The End>