ESP8266

Mesh User Guide

Version 1.0

Espressif Systems IOT Team

http://bbs.espressif.com

Disclaimer and Copyright Notice

Table of Contents

1. Introduction

With the development of the internet of things, there are an increasing number of nodes. However, due to the limited number of nodes that can be directly connected to one router (usually fewer than 64), when the applications have a large number of nodes, it's impossible to have all the nodes directly connected to the same router. There are currently two solutions.

- Super router: we can strengthen the capacity of the routers, so that more nodes can be directly connected to the router.

- Mesh network: the nodes can establish a network and forward the packets.

The ESP8266 uses the mesh network. As a result, a large number of nodes can be connected to the internet without any improvements of the current router.

1.1. Concepts

IOT Espressif App

IOT Espressif App (hereinafter referred to as the IOT app) is a smartphone application developed by Espressif. It can be used to achieve local and remote control of Wi-Fi devices, including smart lights and smart switches.

ESP Touch

ESP Touch is a technology developed by Espressif. You can use ESP Touch to connect your Wi-Fi devices to the router.

Smart config mode for ESP Touch

The device can be configured by ESP Touch only when it is in the smart config mode. This status is called ESP Touch status. For details of the configuration procedure, refer to *2. Mesh Network*.
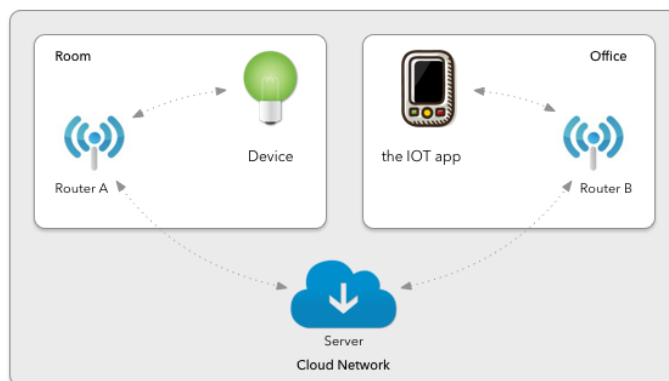
Local device

If a device is configured to the router by ESP Touch, but not activated on the server-end, it is a local device.
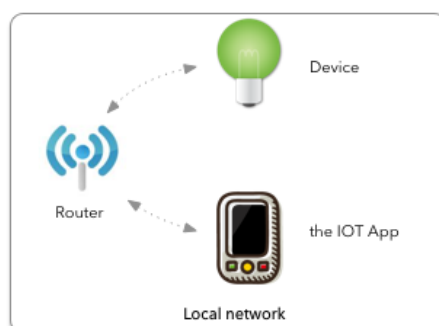
Cloud device

If a device is configured to the router by ESP Touch, and activated on the server-end, it is a cloud device. There are three statuses: cloud status, online status and offline status.

- Cloud status: The device is a cloud device; the device and the IOT app are connected to different routers, as shown in the figure below.

- Online status: The device is a local device or cloud device; the device and the IOT app are connected to the same router, as shown in the figure below.

- Offline status: The device is a cloud device, and is not connected to the router.

Device types and statuses

Device status	Cloud status	Online status	Offline status
Cloud device	✔	✔	✔
Local device	✘	✔	✘

1.2. Nodes

According to the location in a mesh network, nodes can be divided into:

- Root node
    - It receives and sends packets.
    - It forwards packets from the server, smartphone apps and its child nodes.
- Non-root node
    - Non-leaf node: It receives and sends packets, and forwards packets from its parent node and its child nodes.
    - Leaf node: It only receives and sends packets, but does not forward packets.

1.3. Network mode

1.3.1. Network principle

Mesh network supports auto-networking. When ESP Touch is used to configure the mesh network, the device automatically scans the Wi-Fi AP nearby.

The device will start with devices whose RSSI are higher than -45 dbm first. Among these devices, the device will always try to connect to the device with the fewest hop(s) away from the router. It will repeat this process until it has joined the mesh network.

If the device fails to do so, it will choose the devices whose RSSI are lower than -45 dbm, and among these devices, it will always try to connect to the device with the maximum RSSI. It will repeat this process until it has joined the mesh network.

1.3.2. Network diagram

The mesh network diagram is shown as below.
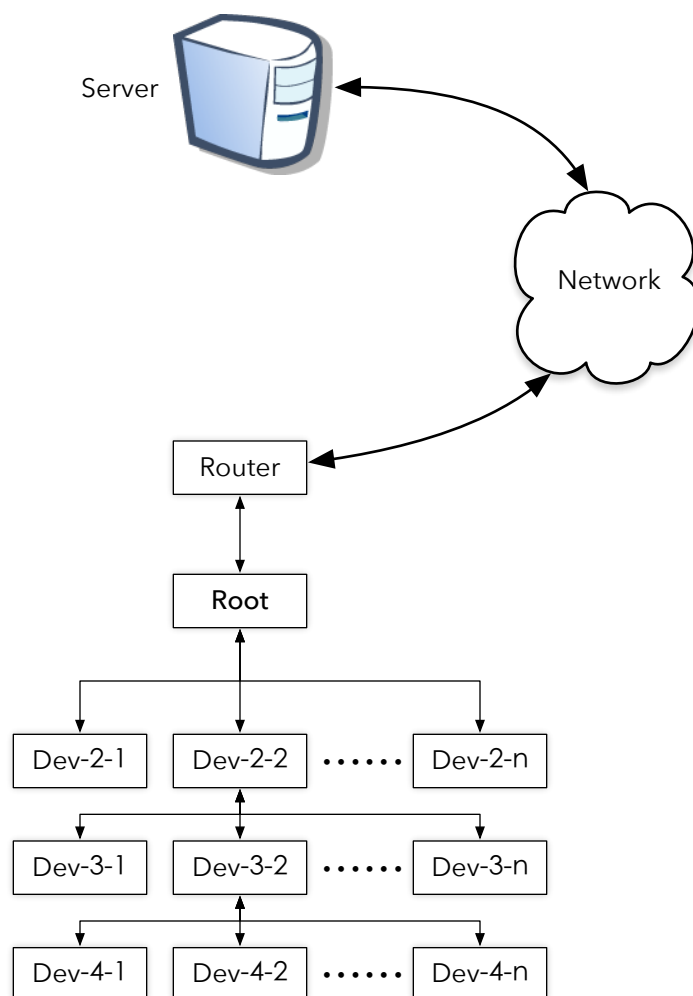
- The node that is directly connected to the router is the root node; the others are non-root nodes.

- When the router is connected to the server, you can use the smartphone to control the cloud devices in the server; otherwise, you can only control the local devices.

2. Configure the mesh network

2.1. Operation process

You can use the ESP Touch function of the IOT app to configure the mesh device. The operation process is shown in the figure below.
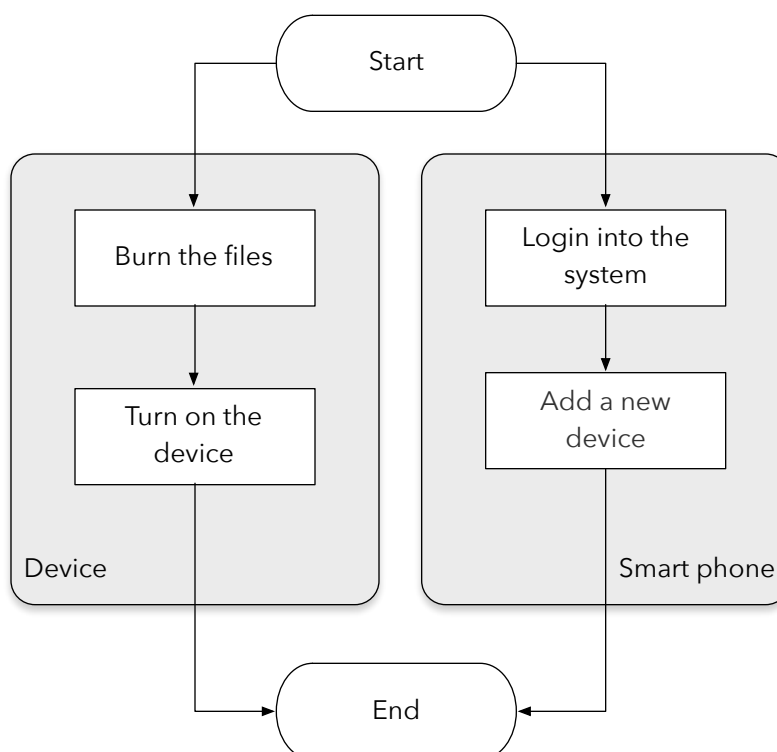
2.2. Hardware requirements

The following hardware devices needed.

Device-end
- A router that can be connected to the internet (If you only need to operate the local devices, you don't need to connect to the internet.)
- Devices with Wi-Fi modules, e.g: smart lights.

Smart phone-end
- A smartphone with the IOT app.

2.3.   Burn the files

> **Note：**
>
> For details of what and how to burn, refer to the *ESP8266__IOT_SDK_User Guide*.

2.4.   Add a new device

> **Note：**
>
> For details of how to login into the system, refer to the *ESP8266_IOT Espressif User Guide*. This section will focus on how to add a new device to the mesh network with ESP Touch.

You can use ESP Touch to set the device as a local or cloud device after the files have been burnt to the device.

Instruction

Add a device named ESP_A10666 to the cloud mesh network with the router whose SSID is ESP_IOE.

> **Note：**
>
> Currently, this system only supports smart lights. Other devices can be supported in the next version of the system.

Requirements

- You need a router whose SSID is ESP_IOE, and can be connected to the internet server.

- Related files have been burnt to Device ESP_A10666, and the device supports IOT Espressif system and the mesh network.
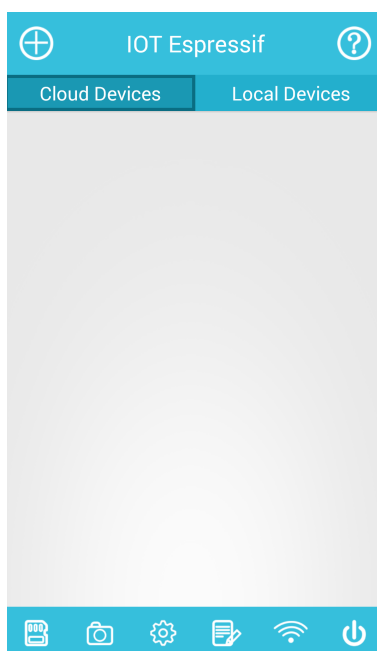
Procedures

1. Touch the IOT Espressif App on the main screen to enter the login page.
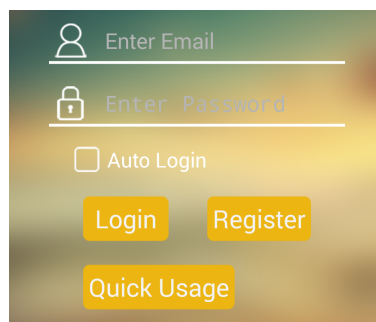
- If you are a new user, touch *Register* to register a new account.

- You can touch *Quick Usage* to operate the local devices.

2. Login into the system with your account and password. The system shows a cloud device list and a local device list.

If you are a new user, the lists are empty.

3. Turn on the device. After the white light is on for 5 seconds, the green light will blink, which means the device can be configured by ESP Touch.

Different colours of the smart light indicate different statuses of the device, as shown in the table below.

| No.	Light colour	device status
1	White (constant on)	The device is on.
2	Green (blinking)	The device can be configured by ESP Touch.
3	Blue (blinking)	A device is added by the IOT app, and is being configured by ESP Touch.
4	White (blinking)	The device has been successfully configured by ESP Touch and connected to the router.
5	Red (blinking)	The device hasn't been configured, or has failed to be configured.

4. Touch the button ✚ to enter the *Add devices* page, and then enter the SSID and the password.

Note:

The system will remember the password, so you don't need to enter the password again for the same Wi-Fi SSID if you have entered it before. Make sure the smartphone is connected to the Wi-Fi network ,or you can not add devices.

SSID	This is the Wi-Fi SSID of the smartphone. You can change it in settings of your smartphone.
Show password	Choose this item to check the password you have entered.
This Wi-Fi is hidden	Choose this item if the Wi-Fi is hidden. Most of the Wi-Fis are not hidden.
Activate device	Choose this item so that you can configure the device and activate it on the server-end. • If you want to configure a local device, don't choose *Activate device*. Configuration takes about 1 minute. • If you want to configure a cloud device, choose *Activate device*. Configuration takes about 1 to 2 minutes.

5. Touch *OK*, and the system shows *Configuring....* When the configuration is completed, the system shows the device lists.
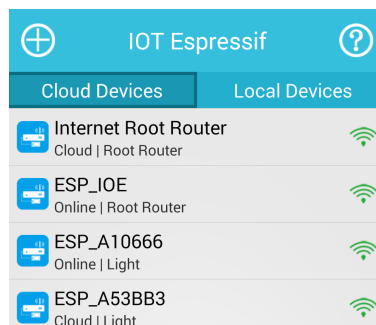
> ⚠**Notice**：
>
> The device can only be connected to the mesh network when it's in the smart config mode for ESP Touch. Don't touch *OK* until the green light is blinking; otherwise, the configuration might fail.

| Internet Root Router	It shows all the cloud devices that are of cloud status or online status.
ESP_IOE	The router SSID that the smartphone is connected to. It shows all the cloud devices of online status.
ESP_A10666	The online device that has been successfully configured this time.
ESP_A53BB3	The cloud devices that have been successfully configured in other routers before.

Tips：

- Hold and slide down the screen to refresh the device lists when the system can't auto-refresh.
- If the configuration is completed (at least one device is connected to the mesh network), the system shows *Configuration completed*. Your device will be in the cloud device list or local device list.
- If the configuration failed, (no device is connected to the mesh network), the system shows *Configuration failed*. Your device will not be in the cloud device list or local device list. You will have to try again.

6. Touch *Internet Root Router* in the device list, and you can see all the cloud devices, including cloud status devices and online status devices.
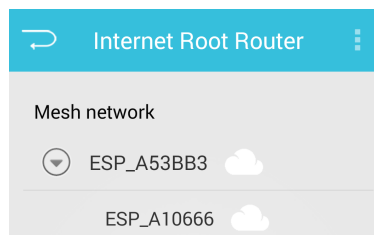
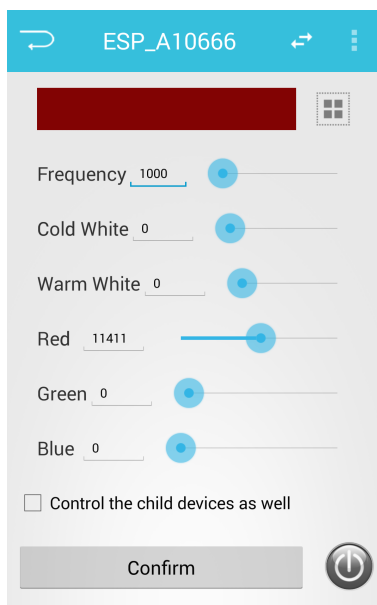
7. Touch *ESP_IOE* in the device list to see all the node devices that are connected to the router.
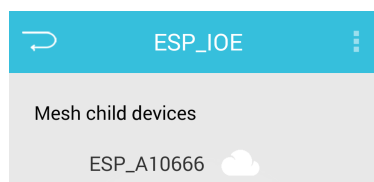
You can see and control the node devices. The device that is directly connected to the router is the root node device, the others are non-root devices.
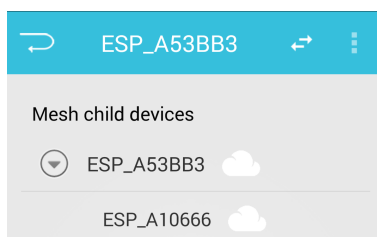
8. Touch *ESP_A10666* to see the operation page of the device.

If you choose *Control the child devices as well*, you can control the device and its child devices at the same time.

9. Touch 🔁 in the ESP_A53BB3 operation page to see the network structure of the device.

-   The device and all its child devices are shown by default.
-   Touch 🔁 to go back to the ESP_A53BB3 operation page.

3.　Software interface

3.1.　Data structure

Data structure of the mesh network is shown as below.

```
typedef void (*espconn_mesh_callback)();
enum mesh_type {
    Mesh_CLOSE = 0,
     Mesh_LOCAL,
     Mesh_ONLINE,
    Mesh_NONE = 0xFF
};
```

3.2.　Interfaces

3.2.1.　espconn_mesh_enable

Function

Enable mesh.

Functional definition

```
void espconn_mesh_enable(espconn_mesh_callback enable_cb, enum mesh_type
type)
```

Parameters

enable_cb	The system will call `enable_cb` when mesh is enabled.
type	Types of mesh; currently, there are two types of mesh: • Mesh_LOCAL • Mesh_ONLINE

> **Note**：
>
> When `espconn_mesh_enable` is called, users should wait for the system to call `anable_cb`, and make subsequent requests in `enable_cb`.

Return

null

3.2.2.   espconn_mesh_disable

Function

Disable mesh.

Functional definition

```
void espconn_mesh_disable(espconn_mesh_callback disable_cb)
```

Parameter

disable_cb	The system will call disable_cb when mesh is disabled.

Return

null

3.2.3.   espconn_mesh_get_status

Function

Get the current status of the mesh network.

Functional definition

```
int8_t espconn_mesh_get_status();
```

Parameter

Return

Mesh_DISABLE	Mesh is disabled.
Mesh_WIFI_CONN	The mesh node is trying to connect to the WiFi.
Mesh_NET_CONN	The mesh node has successfully connected to the WiFi, and is trying to establish a TCP connection.
Mesh_LOCAL_AVAIL	The node has joined the local mesh network.
Mesh_ONLINE_AVAIL	The node has joined the cloud mesh network.