

ESP-NOW

User Manual



Version 1.0

Espressif IoT Team

Copyright © 2015

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 WiFi Alliance Member Logo is a trademark of the WiFi Alliance.

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

Copyright © 2015 Espressif Systems. All rights reserved.

Table of Contents

- 1. Vendor-specific Element1
 - 1.1. Format of Vendor-specific Element.....1
 - 1.2. Content of Vendor-specific Information Element1
- 2. ESP-NOW1
 - Demo code:1



1. Vendor-specific Element

Vendor-specific information element is a set of information elements defined using a unique format and interpreted in a certain way, not described in the IEEE 802.11 standard.

1.1. Format of Vendor-specific Element

The format of Vendor-Specific element is shown in table 1-1 below.

Table 1-1. Format of Vendor-specific Element

	Element ID	Length	OUI	Vendor-specific Content
Length (byte)	1	1	j	n - j
Element ID	Element ID of vendor-specific information, the length of which is 1 byte, and the value is 221 (0xdd).			
Length	Length of "vendor-specific" information elements, the length of which is 1 byte, and the value is n ($j \leq n \leq 255$), n is the total length of OUI and VSC.			
* OUI	Organization Unique Identifier is a 24-bit number that uniquely identifies a vendor, the length of which is $j \geq 3$ byte.			
Vendor-Specific Content	Content of "vendor-specific" information element, the length of the content is $n - j \leq 252$ byte.			

Remark:

* Espressif has reserved the first three octets of MAC address as its OUI, the value is 0x18fe34.

1.2. Content of Vendor-specific Information Element

Content of vendor-specific information element is shown in table 1-2 below.

Table 1-2. Format of the content of vendor-specific information element

	Type	Version	Body
Length (byte)	1	1	n-5



*Type	Type of vendor-specific information element, the length is 1 byte.
Version	Version information of vendor-specific information element, the length is 1 byte.
Body	Content of vendor-specific information element, the length is n-5 ($5 \leq n \leq 255$), the value varies with the changing of type of IE.



2.

ESP-NOW

ESP-NOW is a wireless communication technology built on Action frame that enables encrypted data transmission between devices while no WiFi connection is not required. ESP-NOW technology can be applied to low power consumption devices including various lights, sensors, and so on. The definition of vendor specific body is shown in Figure 2-1 below.

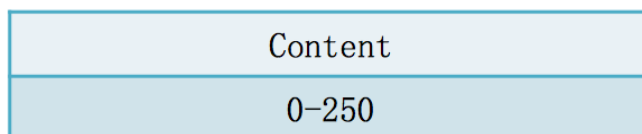


Figure 2-1 ESP-NOW Information Format

Information of ESP-NOW is completely defined by application layer. Information of each frame is entirely transparent to application layer, therefore, only information included in the Content part will be transmitted and received. In this way, the application layer can establish protocols directly so as to control different devices.

Demo code:

```
void ICACHE_FLASH_ATTR simple_cb(u8 *macaddr, u8 *data, u8 len)
{
    int i;
    u8 ack_buf[16];
    u8 recv_buf[17];

    os_printf("now from");
    for (i = 0; i < 6; i++)
        os_printf("%02X, ", macaddr[i]);
    os_printf(" len: %d]:", len);

    os_bzero(recv_buf, 17);
    os_memcpy(recv_buf, data, len<17?len:16);

    if (os_strncmp(data, "ACK", 3) == 0)
```



```
        return;

    os_sprintf(ack_buf, "ACK[%08x]", ack_count++);
    esp_now_send(macaddr, ack_buf, os_strlen(ack_buf));
}

void user_init(void)
{
    u8 key[16]= {0x33, 0x44, 0x33, 0x44, 0x33, 0x44, 0x33, 0x44, 0x33, 0x44,
0x33, 0x44, 0x33, 0x44, 0x33, 0x44};
    u8 da1[6] = {0x18, 0xfe, 0x34, 0x97, 0xd5, 0xb1};
    u8 da2[6] = {0x1a, 0xfe, 0x34, 0x97, 0xd5, 0xb1};

    if (esp_now_init()==0) {
        os_printf("esp_now init ok\n");

        esp_now_register_recv_cb(simple_cb);
        esp_now_set_self_role(1);
        esp_now_add_peer(da1, 1, key, 16);
        esp_now_add_peer(da2, 2, key, 16)

    } else {
        os_printf("esp_now init failed\n");
    }
}

void ICACHE_FLASH_ATTR demo_send(u8 *mac_addr, u8 *data, u8 len)
{
    esp_now_send(NULL, data, len);/* the demo will send to two devices which
added by esp_now_add_peer() */
    //esp_now_send(mac_addr, data, len); /* send to the specified mac_addr */
}
```