

# ESP-NOW

## 用户手册



版本 1.0

乐鑫科技 IOT 团队

<http://bbs.espressif.com>

Copyright © 2015

## **免责声明和版权公告**

本文中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。

文档“按现状”提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

Wi-Fi 联盟成员标志归 Wi-Fi 联盟所有。

文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

版权归© 2015 乐鑫信息科技（上海）有限公司所有。保留所有权利。

# 目录

1.	Vendor-Specific 元素.....	1
1.1.	Vendor-Specific 元素的格式.....	1
1.2.	Vendor-Specific 内容.....	1
2.	ESP-NOW .....	3
	Demo code.....	3



# 1. Vendor-Specific 元素

在 IEEE 802.11 标准中，Vendor-Specific 元素使用一种单独的格式来定义，用于承载没有在 IEEE 802.11 标准中定义的信息。

## 1.1. Vendor-Specific 元素的格式

Vendor-Specific 元素的格式如表 1-1 所示。

表 1-1. Vendor-Specific 元素的格式

	Element ID	Length	OUI	Vendor-Specific Content
长度 (字节)	1	1	j	n - j
Element ID	Vendor-Specific 元素的 ID，长度为 1 字节，其值为 221 (0xdd)。			
Length	Vendor-Specific 元素的长度，长度为 1 字节，其值为 n ( $j \leq n \leq 255$ )，n 是 OUI 和 VSC 的长度总和。			
*OUI	Organization Unique Identifier 组织唯一标识符，长度为 $j \geq 3$ 字节。			
Vendor-Specific Content	Vendor-Specific 的内容，长度为 $n - j \leq 252$ 字节。			

### 说明：

\* 乐鑫已选用其申请的第一个 MAC 地址段作为其 OUI，其值为 0x18fe34。

## 1.2. Vendor-Specific 内容

Vendor-Specific 内容的格式如表 1-2 所示。

表 1-2. Vendor-Specific 内容的格式

	Type	Version	Body
长度 (字节)	1	1	n-5
*Type	Vendor-Specific 元素的类型，长度为 1 字节。		



---

<b>Version</b>	Vendor-Specific 元素的版本号，长度为 1 字节。
<b>Body</b>	Vendor-Specific 元素的内容，长度为 n-5 字节。 $5 \leq n \leq 255$ ，会根据不同的 IE 类型而变动。



## 2.

# ESP-NOW

ESP-NOW 是基于 ACTION 帧的无线传输技术。ESP-NOW不需要建立 Wi-Fi 连接，可以在设备间直接传输可加密数据信息，适用于控制各种灯光、传感器等低功耗设备。Vendor Specifier Body 定义如图 2-1。

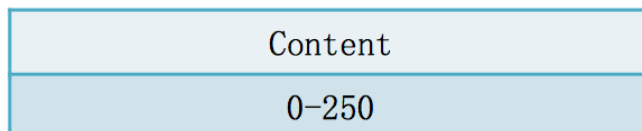


图 2-1 ESP-NOW Information Format

ESP-NOW 的信息内容完全由应用层定义，各种帧的封装对应用层是透明的，应用层只需要填 Content 部分，就可以发送；接受时，应用层收到的只有 Content 的部分。这样应用层可以直接自组协议,用于控制各种设备。

### 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 */
}
```