

第一章 新建一个 WIFI 热点

1. 学习目的及目标

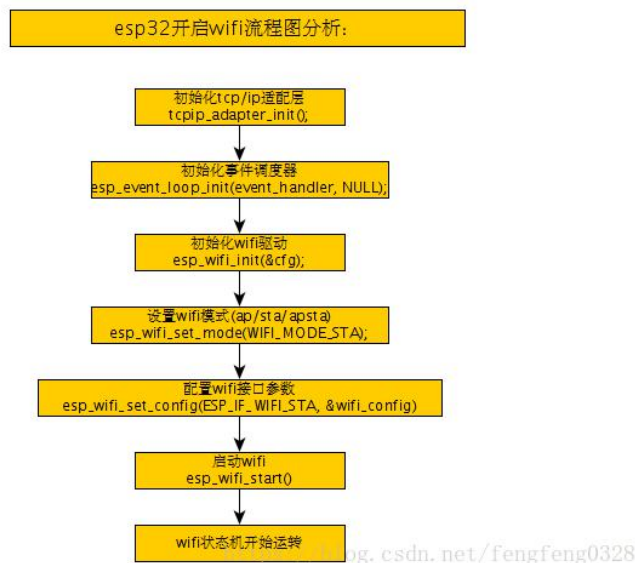
- 掌握 WIFI 的状态机结构
- 掌握乐鑫新建热点（AP）的程序

2. ESP32 的 WiFi 特点

- WiFi 可作为 AP、STA、AP+STA 三种模式
- 作为 AP 时，可以被 4 个 STA 连接
- AP 是 (Wireless) AccessPoint 的缩写，即（无线）访问接入点。简单来讲就像是无线路由器一样，设备打开后进入 AP 模式，在手机的网络列表里面，可以搜索到类似 TPLINK_XXX 的名字（SSID）
- STA 是每一个连接到无线网络中的终端（如笔记本电脑、手机、PAD 及其它可以联网的用户设备）都可称为一个站点，STA 本身并不接受无线的接入，它可以连接到 AP。

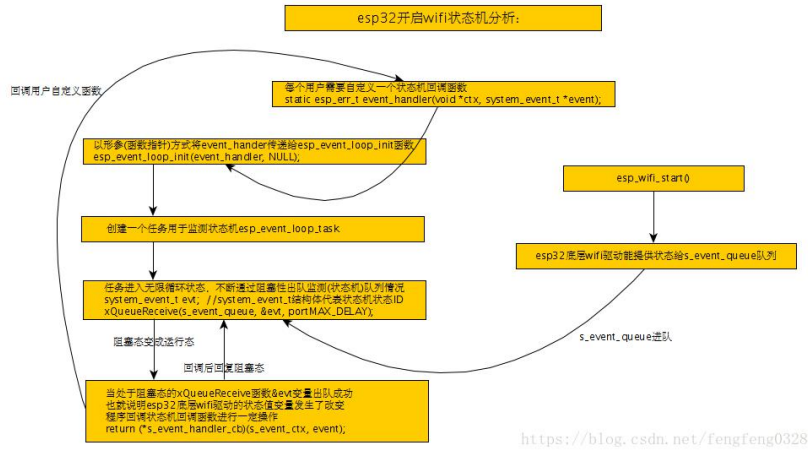
3. 软件设计

3.1. 代码逻辑（[来自网络](#)）



3.2. ESP32 的 I2C master 接口介绍（[来自网络](#)）

esp32wifi 底层驱动是不开源的，以下是 esp32 wifi 状态机工作原理的流程图



3.3. ESP32 的 WIFI 状态机总结

初始化 wifi 配置后，程序会根据 WIFI 的实时状态，在回调函数中给出状态返回，所以只需要在回调中进行相关操作，接下来讲解状态机的解析。

3.4. 状态机解析

回调原型	<pre>static esp_err_t event_handler (void *ctx, system_event_t *event)</pre>
函数功能	WIFI 状态机回调函数
参数	<pre>[in] event: typedef struct { system_event_id_t event_id; /*事件 ID*/ system_event_info_t event_info; /*事件信息*/ } system_event_t; //事件 ID 取值 typedef enum { SYSTEM_EVENT_WIFI_READY = 0, /*WiFi 准备好*/ SYSTEM_EVENT_SCAN_DONE, /*扫描 AP 完成*/ SYSTEM_EVENT_STA_START, /*作为 STA 开始工作*/ SYSTEM_EVENT_STA_STOP, /*作为 STA 结束工作*/ SYSTEM_EVENT_STA_CONNECTED, /*作为 STA 连接上 AP*/ SYSTEM_EVENT_STA_DISCONNECTED, /*作为 STA 断开 AP*/ SYSTEM_EVENT_STA_AUTHMODE_CHANGE, /*作为 STA 连接的 AP 改变了*/ SYSTEM_EVENT_STA_GOT_IP, /*作为 STA 从 AP 获取了 IP*/ SYSTEM_EVENT_STA_LOST_IP, /*作为 STA 丢掉了 IP, 也就是错了*/ SYSTEM_EVENT_STA_WPS_ER_SUCCESS, /*wps succeeds in enrollee mode*/ SYSTEM_EVENT_STA_WPS_ER_FAILED, /*wps fails in enrollee mode*/ SYSTEM_EVENT_STA_WPS_ER_TIMEOUT, /*wps timeout in enrollee mode*/ SYSTEM_EVENT_STA_WPS_ER_PIN, /*wps pin code in enrollee mode*/ SYSTEM_EVENT_AP_START, /*AP 开始*/ SYSTEM_EVENT_AP_STOP, /*AP 结束*/ SYSTEM_EVENT_AP_STACONNECTED, /*有 STA 连上 SP32 的 AP*/</pre>

	<pre>SYSTEM_EVENT_AP_STADISCONNECTED, /*有 STA 断开 ESP32 的 AP*/ SYSTEM_EVENT_AP_PROBEREQRECVED, /*从 AP 中接收包*/ SYSTEM_EVENT_GOT_IP6, /*获取 IPv6*/ SYSTEM_EVENT_ETH_START, /*有线开始*/ SYSTEM_EVENT_ETH_STOP, /*有线停止*/ SYSTEM_EVENT_ETH_CONNECTED, /*有线上线*/ SYSTEM_EVENT_ETH_DISCONNECTED, /*有线掉线*/ SYSTEM_EVENT_ETH_GOT_IP, /*有线获取了 IP*/ SYSTEM_EVENT_MAX } system_event_id_t;</pre>
返回值	ESP_OK : 成功 ESP_ERR_INVALID_ARG : 参数错误

3.5. WiFi 初始化 API 介绍

➤ 设置 WiFi 回调函数 `esp_event_loop_init()`;

回调原型	<pre>esp_err_t esp_event_loop_init ()</pre>
函数功能	设置 WiFi 回调函数
参数	[in] cb:回调函数指针 [in] ctx:保留
返回值	ESP_OK : 成功 other: 失败

➤ 初始化 WiFi 设置函数 `WIFI_INIT_CONFIG_DEFAULT()`;

➤ 设置 WiFi 工作模式函数 `esp_err_t esp_wifi_set_mode()`;

回调原型	<pre>esp_err_t esp_wifi_set_mode (wifi_mode_t mode)</pre>
函数功能	设置 WiFi 工作模式函数
参数	[in] mode:wifi 工作模式
返回值	ESP_OK : 成功 other: 失败

➤ 启动 WiFi 函数: `esp_err_t esp_wifi_start(void)`;

更多更详细接口请参考[官方指南](#)。

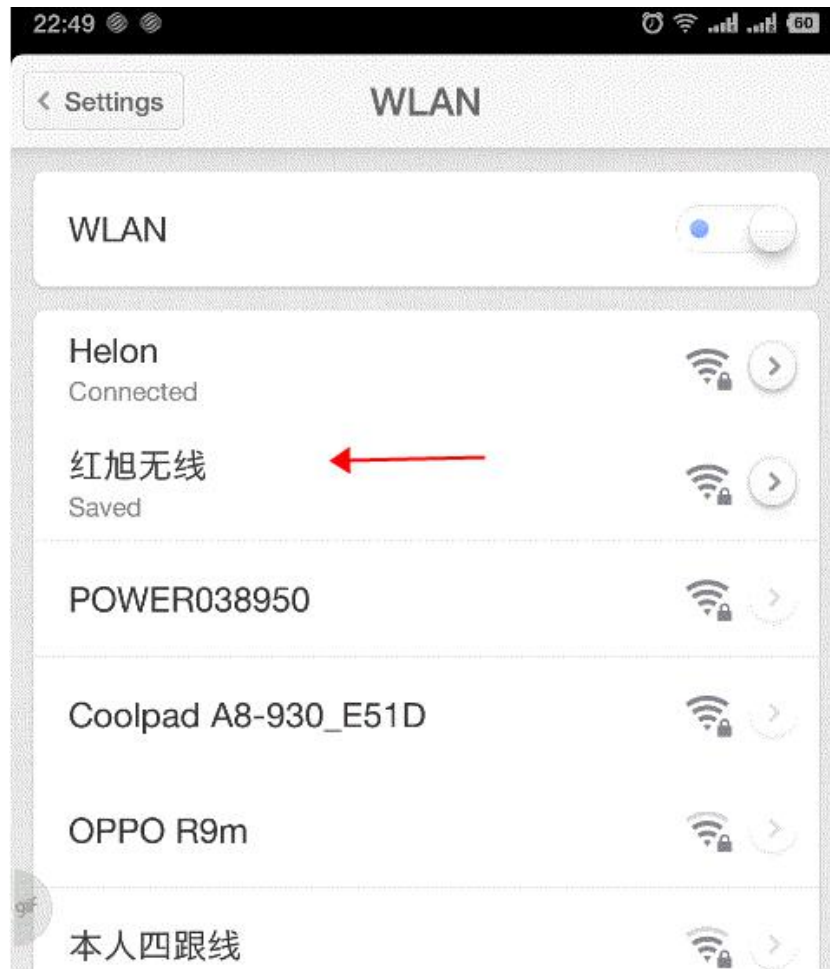
3.6. WiFi 初始化编写

```
1 void app_main()
2 {
3     ESP_ERROR_CHECK( nvs_flash_init() );
4     tcpip_adapter_init();
5     ESP_ERROR_CHECK(esp_event_loop_init(event_handler, NULL));
6
7     wifi_init_config_t cfg = WIFI_INIT_CONFIG_DEFAULT();
8     ESP_ERROR_CHECK(esp_wifi_init(&cfg));
9     wifi_config_t wifi_config = {
10         .ap = {
11             .ssid = "Helon_test",
12             .ssid_len = 0,
13             /* 最多只能被 4 个 station 同时连接,这里设置为只能被一个 station 连接 */
14             .max_connection = 1,
15             .password = "20180604",
16             .authmode = WIFI_AUTH_WPA_WPA2_PSK,
17         },
18     };
19     ESP_ERROR_CHECK(esp_wifi_set_mode(WIFI_MODE_AP));
20     ESP_ERROR_CHECK(esp_wifi_set_config(ESP_IF_WIFI_AP, &wifi_config));
21     ESP_ERROR_CHECK(esp_wifi_start());
22 }
```

3.7. 作为 AP 的回调函数解析

```
1 static esp_err_t event_handler(void *ctx, system_event_t *event)
2 {
3     switch (event->event_id)
4     {
5         case SYSTEM_EVENT_AP_START:
6             printf("\nwifi_softap_start\n");
7             break;
8         case SYSTEM_EVENT_AP_STA_CONNECTED:
9             printf("\nwifi_softap_connected\n");
10            break;
11         case SYSTEM_EVENT_AP_STA_DISCONNECTED:
12             printf("\nwifi_softap_disconnected\n");
13             break;
14         default:
15             break;
16     }
17     return ESP_OK;
```

4. 效果展示



5. WiFi 总结

- 乐鑫采用了大众的，初始化+回调的状态机结构，逻辑清晰简单，方便学习和使用。
- 源码地址: <https://github.com/xiaolongba/wireless-tech>