

WIFI STATION

```
#include <string.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/event_groups.h"
#include "esp_system.h"
#include "esp_wifi.h"
#include "esp_event.h"
#include "esp_log.h"
#include "nvs_flash.h"

#include "lwip/err.h"
#include "lwip/sys.h"

#define EXAMPLE_ESP_WIFI_SSID CONFIG_ESP_WIFI_SSID
#define EXAMPLE_ESP_WIFI_PASS CONFIG_ESP_WIFI_PASSWORD
#define EXAMPLE_ESP_MAXIMUM_RETRY CONFIG_ESP_MAXIMUM_RETRY

static EventGroupHandle_t s_wifi_event_group;
const int WIFI_CONNECTED_BIT = BIT0;
static const char *TAG = "wifi station";
static int s_retry_num = 0;

static void event_handler(void *arg, esp_event_base_t event_base,
int32_t event_id, void **event_data) {
    if (event_base == WIFI_EVENT && event_id == WIFI_EVENT_STA_START) {
        esp_wifi_connect();
    }
```

```

    } else if (event_base == WIFI_EVENT_STA_DISCONNECTED) {
        if (s_retry_num < EXAMPLE_ESP_MAXIMUM_RETRY) {
            esp_wifi_connect();
            xEventGroupClearBits(s_wifi_event_group, WIFI_CONNECTED_BIT);
            s_retry_num++;
            ESP_LOGI(TAG, "retry to connect to the AP");
        }
        ESP_LOGI(TAG, "connect to the AP fail");
    } else if (event_base == IP_EVENT && event_id == IP_EVENT_STA_GOT_IP) {
        ip_event_got_ip_t *event = (ip_event_got_ip_t *) event_data;
        ESP_LOGI(TAG, "got ip: %s", ipaddr_ntoa(event->ip_info.ip));
        s_retry_num = 0;
        xEventGroupSetBits(s_wifi_event_group, WIFI_CONNECTED_BIT);
    }
}

```

```

void wifi_init_sta() {
    s_wifi_event_group = xEventGroupCreate();
    tcpip_adapter_init();
    ESP_ERROR_CHECK(esp_event_loop_create_default());
    wifi_init_config_t cfg = WIFI_INIT_CONFIG_DEFAULT();
    ESP_ERROR_CHECK(esp_wifi_init(&cfg));
    ESP_ERROR_CHECK(esp_event_handler_register(
        WIFI_EVENT, ESP_EVENT_ANY_ID, &event_handler, NULL));
    ESP_ERROR_CHECK(esp_event_handler_register(
        IP_EVENT, IP_EVENT_STA_GOT_IP, &event_handler, NULL));

    wifi_config_t wifi_config = {

```

```
    .sta = 1
```

```
    .ssid = EXAMPLE_ESP_WIFI_SSID,
```

```
    .password = EXAMPLE_ESP_WIFI_PASSWORD, &wifi_config)),
```

```
    }
```

```
    },
```

```
    ESP_ERROR_CHECK(esp_wifi_set_mode(WIFI_MODE_STA));
```

```
    ESP_ERROR_CHECK(esp_wifi_set_config(ESP_IF_WIFI_STA, &wifi_config));
```

```
    ESP_ERROR_CHECK(esp_wifi_start());
```

```
    ESP_LOGI(TAG, "wifi init sta finished.");
```

```
    ESP_LOGI(TAG, "connect to AP SSID: %s password: %s",
```

```
        EXAMPLE_ESP_WIFI_SSID, EXAMPLE_ESP_WIFI_PASS);
```

```
    }
```

```
void app_main() {
```

```
    esp_err_t ret = nvs_flash_init();
```

```
    if (ret == ESP_ERR_NVS_NO_FREE_PAGES ||
```

```
        ret == ESP_ERR_NVS_NEW_VERSION_FOUND) {
```

```
        ESP_ERROR_CHECK(nvs_flash_erase());
```

```
        ret = nvs_flash_init();
```

```
    }
```

```
    ESP_ERROR_CHECK(ret);
```

```
    ESP_LOGI(TAG, "ESP_WIFI_MODE_STA");
```

```
    wifi_init_sta();
```

```
}
```