



**DesignNews**

Designing Embedded Systems using the ESP32

## DAY 4 : It's all about Wi-Fi

Sponsored by



## Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click "Help" or submit a question asking for assistance.
- Participate in 'Group Chat' by maximizing the chat widget in your dock.

## Course Sessions

- Introduction to the ESP32 Wi-Fi Module
- Setting up and Exploring the SDK
- Programming and Writing the First Application
- **It's all about Wi-Fi**
- Jump-Starting Cloud Connectivity Applications with Amazon FreeRTOS

## A closer look at Hello World

```
void app_main(void)
{
    printf("Hello world!\n");

    /* Print chip information */
    esp_chip_info_t chip_info;
    esp_chip_info(&chip_info);
    printf("This is %s chip with %d CPU cores, WiFi%s%s, ",
        CONFIG_IDF_TARGET,
        chip_info.cores,
        (chip_info.features & CHIP_FEATURE_BT) ? "/BT" : "",
        (chip_info.features & CHIP_FEATURE_BLE) ? "/BLE" : "");

    printf("silicon revision %d, ", chip_info.revision);

    printf("%dMB %s flash\n", spi_flash_get_chip_size() / (1024 * 1024),
        (chip_info.features & CHIP_FEATURE_EMB_FLASH) ? "embedded" : "external");

    printf("Free heap: %d\n", esp_get_free_heap_size());

    for (int i = 10; i >= 0; i--) {
        printf("Restarting in %d seconds...\n", i);
        vTaskDelay(1000 / portTICK_PERIOD_MS);
    }

    printf("Restarting now.\n");
    fflush(stdout);
    esp_restart();
}
```

## Adding Blinky support

```
void app_main(void)
{
    xTaskCreate(&blinky, "blinky", 512, NULL, 5, NULL );
    xTaskCreate(&helloWorld, "hello", 2048, NULL, 5, NULL );

    /* Print chip information */
    esp_chip_info_t chip_info;
    esp_chip_info(&chip_info);
    printf("This is %s chip with %d CPU cores, WiFi%s%s, ",
        CONFIG_IDF_TARGET,
        chip_info.cores,
        (chip_info.features & CHIP_FEATURE_BT) ? "/BT" : "",
        (chip_info.features & CHIP_FEATURE_BLE) ? "/BLE" : "");

    printf("silicon revision %d, ", chip_info.revision);

    printf("%dMB %s flash\n", spi_flash_get_chip_size() / (1024 * 1024),
        (chip_info.features & CHIP_FEATURE_EMB_FLASH) ? "embedded" : "external");

    printf("Free heap: %d\n", esp_get_free_heap_size());

    while(1)
    {
        vTaskDelay(1000 / portTICK_PERIOD_MS);
    }
}
```

## Adding Blinky support

```
#include <stdio.h>
#include "sdkconfig.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_system.h"
#include "esp_spi_flash.h"
#include "driver/gpio.h"
```

```
#define LED_RED 0
#define LED_GREEN 2
#define LED_BLUE 4
```

## Adding Blinky Support

```
void blinky(void *pvParameter)
```

```
{
```

```
    gpio_pad_select_gpio(LED_RED);  
    /* Set the GPIO as a push/pull output */  
    gpio_set_direction(LED_RED, GPIO_MODE_OUTPUT);  
  
    gpio_pad_select_gpio(LED_GREEN);  
    /* Set the GPIO as a push/pull output */  
    gpio_set_direction(LED_GREEN, GPIO_MODE_OUTPUT);  
  
    gpio_pad_select_gpio(LED_BLUE);  
    /* Set the GPIO as a push/pull output */  
    gpio_set_direction(LED_BLUE, GPIO_MODE_OUTPUT);
```

```
while(1) {
```

```
    /* Blink off (output low) */  
    gpio_set_level(LED_RED, 1);  
    gpio_set_level(LED_GREEN, 0);  
    gpio_set_level(LED_BLUE, 0);  
  
    vTaskDelay(1000 / portTICK_RATE_MS);  
    /* Blink on (output high) */  
    gpio_set_level(LED_RED, 0);  
    gpio_set_level(LED_GREEN, 1);  
    gpio_set_level(LED_BLUE, 0);  
    vTaskDelay(1000 / portTICK_RATE_MS);  
  
    gpio_set_level(LED_RED, 0);  
    gpio_set_level(LED_GREEN, 0);  
    gpio_set_level(LED_BLUE, 1);  
    vTaskDelay(1000 / portTICK_RATE_MS);
```

```
    }  
}
```

Do you understand the basic organization and model now for adding tasks to ESP32 applications?

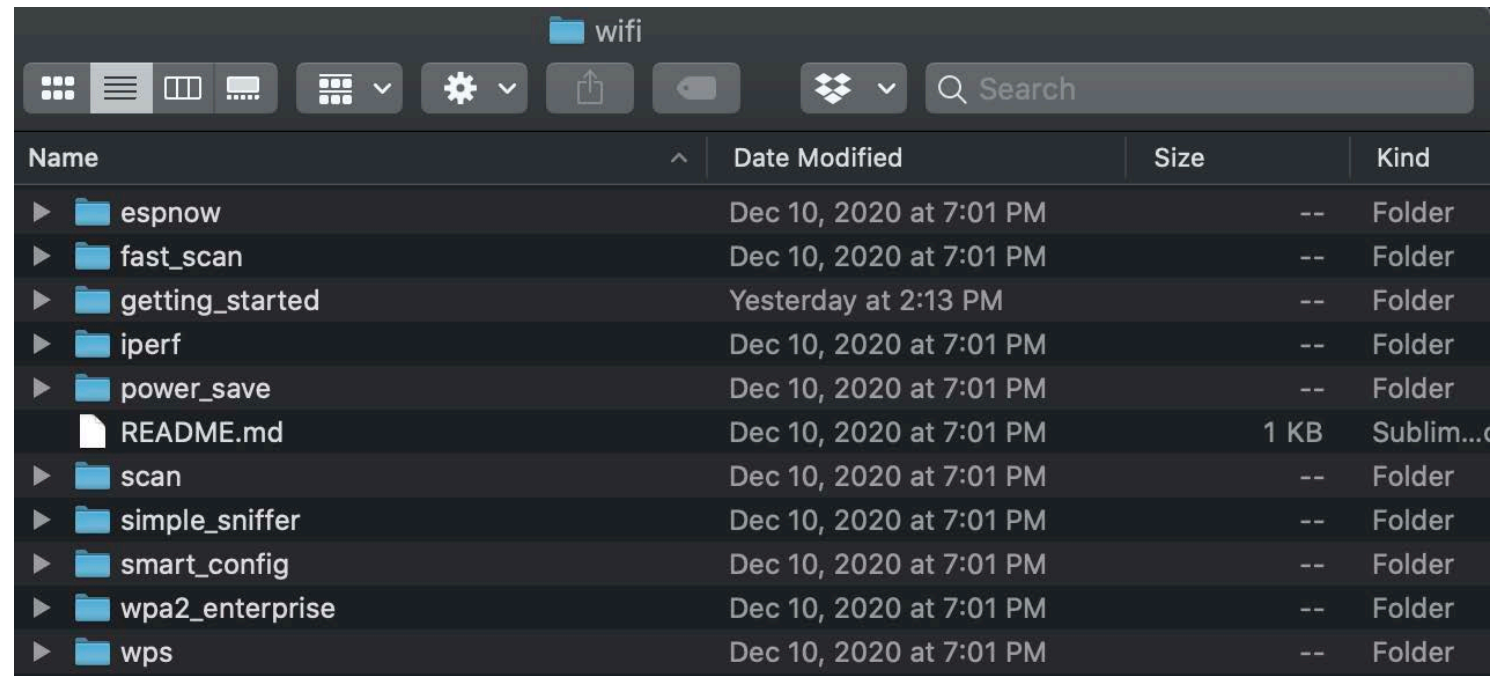
- Yes
- No



## Wi-Fi Examples

esp-idf/examples/wifi/getting\_started

- softAP
  - softap\_example\_main.c
- station
  - station\_example\_main.c



## Wi-Fi Access Point

```
#define EXAMPLE_ESP_WIFI_SSID          CONFIG_ESP_WIFI_SSID
#define EXAMPLE_ESP_WIFI_PASS          CONFIG_ESP_WIFI_PASSWORD
#define EXAMPLE_ESP_WIFI_CHANNEL       CONFIG_ESP_WIFI_CHANNEL
#define EXAMPLE_MAX_STA_CONN           CONFIG_ESP_MAX_STA_CONN
```

Which of the following can be used to change the SSID and password?

- Update the WIFI\_SSID and WIFI\_PASS macros with strings
- Update the project configuration using menuconfig
- All of the above
- None of the above

## Wi-Fi Access Point

```
void app_main(void)
{
    //Initialize NVS
    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_AP");
    wifi_init_softap();
}
```

## Side Note: Application Logging

- ESP\_LOGE – error
- ESP\_LOGW – warning
- ESP\_LOGI – info
- ESP\_LOGD – debug
- ESP\_LOGV - verbose

```
I (0) cpu_start: App cpu up.
I (239) heap_init: Initializing. RAM available for dynamic allocation:
I (246) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (252) heap_init: At 3FFB28C8 len 0002D738 (181 KiB): DRAM
I (259) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (265) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (271) heap_init: At 40089ECC len 00016134 (88 KiB): IRAM
I (278) cpu_start: Pro cpu start user code
I (296) spi_flash: detected chip: gd
I (296) spi_flash: flash io: dio
W (297) spi_flash: Detected size(4096k) larger than the size in the binary image header(2048k)
I (306) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
RHello World 10 !
Hello world!
This is esp32 chip with 2 CPU cores, WiFi/BT/BLE, silicon revision 1, 2MB external flash
Free heap: 296628
E (339) wifi softAP: LOGE Output
W (349) wifi softAP: LOGW Output
I (349) wifi softAP: LOGI Output
```

## Wi-Fi Access Point

Running the example:

- 1) Copy the example
- 2) Navigate into the example folder
- 3) `idf.py build`
- 4) `idf.py -p /dev/cu.usbserial-14301 flash`
- 5) `idf.py -p /dev/cu.usbserial-14301 monitor`

## Wi-Fi Access Point

```
I (499) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
I (588) wifi softAP: ESP_WIFI_MODE_AP
I (598) wifi:wifi driver task: 3ffc05c4, prio:23, stack:6656, core=0
I (598) system_api: Base MAC address is not set
I (598) system_api: read default base MAC address from EFUSE
I (618) wifi:wifi firmware version: 1865b55
I (618) wifi:wifi certification version: v7.0
I (618) wifi:config NVS flash: enabled
I (618) wifi:config nano formatting: disabled
I (628) wifi:Init data frame dynamic rx buffer num: 32
I (628) wifi:Init management frame dynamic rx buffer num: 32
I (638) wifi:Init management short buffer num: 32
I (638) wifi:Init dynamic tx buffer num: 32
I (648) wifi:Init static rx buffer size: 1600
I (648) wifi:Init static rx buffer num: 10
I (648) wifi:Init dynamic rx buffer num: 32
I (658) wifi_init: rx ba win: 6
I (658) wifi_init: tcpip mbox: 32
I (668) wifi_init: udp mbox: 6
I (668) wifi_init: tcp mbox: 6
I (668) wifi_init: tcp tx win: 5744
I (678) wifi_init: tcp rx win: 5744
I (678) wifi_init: tcp mss: 1440
I (688) wifi_init: WiFi IRAM OP enabled
I (688) wifi_init: WiFi RX IRAM OP enabled
I (788) phy: phy_version: 4500, 0cd6843, Sep 17 2020, 15:37:07, 0, 0
I (788) wifi:mode : softAP (3c:71:bf:47:37:09)
I (798) wifi:Total power save buffer number: 16
I (798) wifi:Init max length of beacon: 752/752
I (798) wifi:Init max length of beacon: 752/752
I (798) wifi softAP: wifi_init_softap finished. SSID:myssid password:mypassword channel:1
```

## Wi-Fi Station Mode

```
I (773) wifi:mode : sta (3c:71:bf:47:37:08)
I (773) wifi station: wifi_init_sta finished.
I (2823) wifi station: retry to connect to the AP
I (2823) wifi station: connect to the AP fail
I (4873) wifi station: retry to connect to the AP
I (4873) wifi station: connect to the AP fail
I (6923) wifi station: retry to connect to the AP
I (6923) wifi station: connect to the AP fail
I (8973) wifi station: retry to connect to the AP
I (8973) wifi station: connect to the AP fail
I (11023) wifi station: retry to connect to the AP
I (11023) wifi station: connect to the AP fail
I (13063) wifi station: connect to the AP fail
I (13063) wifi station: Failed to connect to SSID:myssid, password:mypassword
```



## Wi-Fi Station Mode

- 1) idf.py menuconfig
- 2) Example Configuration

```
(Top) > Example Configuration
(myssid) WiFi SSID
(mypassword) WiFi Password
(5) Maximum retry
```

- 3) Update
- 4) Q to exit and then Y to save
- 5) Rebuild and deploy

## Wi-Fi Station Mode

```
I (764) phy: phy_version: 4500, 0cd6843, Sep 17 2020, 15:37:07, 0, 0
I (774) wifi:mode : sta (3c:71:bf:47:37:08)
I (774) wifi station: wifi_init_sta finished.
I (894) wifi:new:<1,0>, old:<1,0>, ap:<255,255>, sta:<1,0>, prof:1
I (894) wifi:state: init -> auth (b0)
I (904) wifi:state: auth -> assoc (0)
I (914) wifi:state: assoc -> run (10)
I (924) wifi:connected with MADSCIE, aid = 4, channel 1, BW20, bssid = 60:5f:8d:d1:85:53
I (924) wifi:security: WPA2-PSK, phy: bgn, rssi: -37
I (934) wifi:pm start, type: 1

I (944) wifi:AP's beacon interval = 102400 us, DTIM period = 2
I (1574) esp_netif_handlers: sta ip: 10.0.0.247, mask: 255.0.0.0, gw: 10.0.0.1
I (1574) wifi station: got ip:10.0.0.247
I (1574) wifi station: connected to ap SSID:MADSCIE
```

Which mode are you most interested in using in your application?

- Access Point mode
- Station Mode
- Dual AP/Station mode

## Next Steps

- Run a socket server
- Connect to a socket server
- TCP Server
- TCP Client
- UDP Server
- UDP Client

Visit [bit.ly/381o9zc](https://bit.ly/381o9zc) to learn more about lwIP support.

## Thank you for attending

Please consider the resources below:

- [www.beningo.com](http://www.beningo.com)
  - Blog, White Papers, Courses
  - Embedded Bytes Newsletter
    - <http://bit.ly/1BAHYXm>

From [www.beningo.com](http://www.beningo.com) under

- Blog > CEC – Designing Embedded Systems using the ESP32





**DesignNews**

Thank You

Sponsored by

