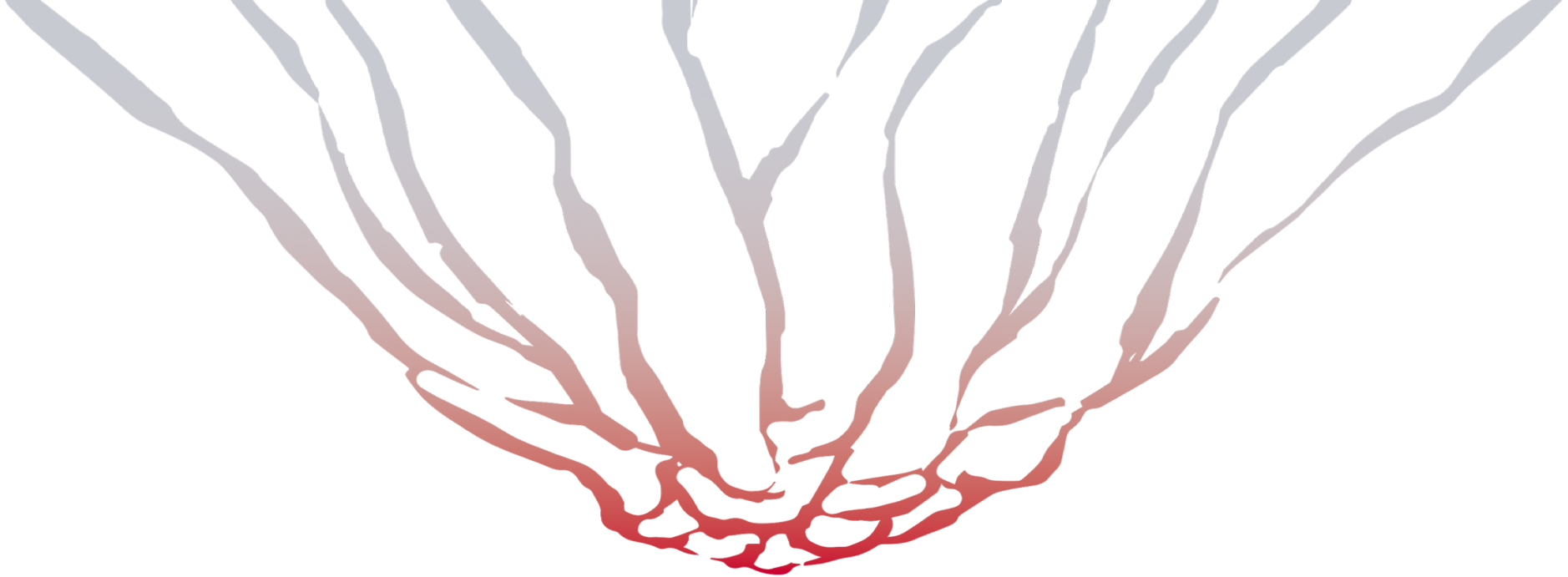


ESPRESSIF

TRAINING_Day 1

ESP Jumpstart

ESP Jumpstart Overview

Getting started

Part 1: Hello World !

Part 2: Driver

Part 3: WIFI

Part 4: Network Configuration

Hands–on session & Demo Show

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Agenda

项目	时间	时长	内容
Opening Ceremony & Espressif Introduction	9:30-10:15	45min	开营仪式 乐鑫公司与产品介绍（人才计划）
Getting started with Jumpstart	10:30-12:00	1.5h	1. JumpStart 项目介绍 2. ESP32 开发环境、SDK（ESP-IDF->MDF、ADF） 介绍、工程结构等 3. JumpStart 与 IDF 关系 4. 搭建 JumpStart 开发环境 5. 从源代码到机器码（执行指令） bin 以及烧写的过程（编译原理浅述）

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Agenda

Hello world!	1:00-2:00	1h	1. ESP32 启动流程 2. FreeRTOS 操作系统简介
Driver	2:10-3:40	1.5h	1. 实体按钮实现 2. 输出端 3. 按键去抖动、继电器控制 4. ESP32 GPIO 资源、使用介绍 5. 扩充: 利用 driver 实现其他的产品形态
WiFi			1. Wi-Fi 介绍（AP、STA、Router） 2. Wi-Fi 编程模型介绍、示意图 3. Wi-Fi 信息硬编码的弊端

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Agenda

Network Configuration	3:50-5:20	1.5h	1. 设备为什么要配网? 2. 网络配置介绍、配网方式介绍（SoftAP、BLE） 3. 演示过程 4. 统一配置 5. 网络信息保存，NVS 介绍 6. 恢复出产设置
Hands-on session + Demo Show	5:30-6:30	1h	1. 人体检测-灯光控制-状态反馈到 PC/手机 2. 温湿度控制器-状态反馈到 PC/手机 3. Espressif Show: ESP32-Touch-LCD, ESP-Sense Kit

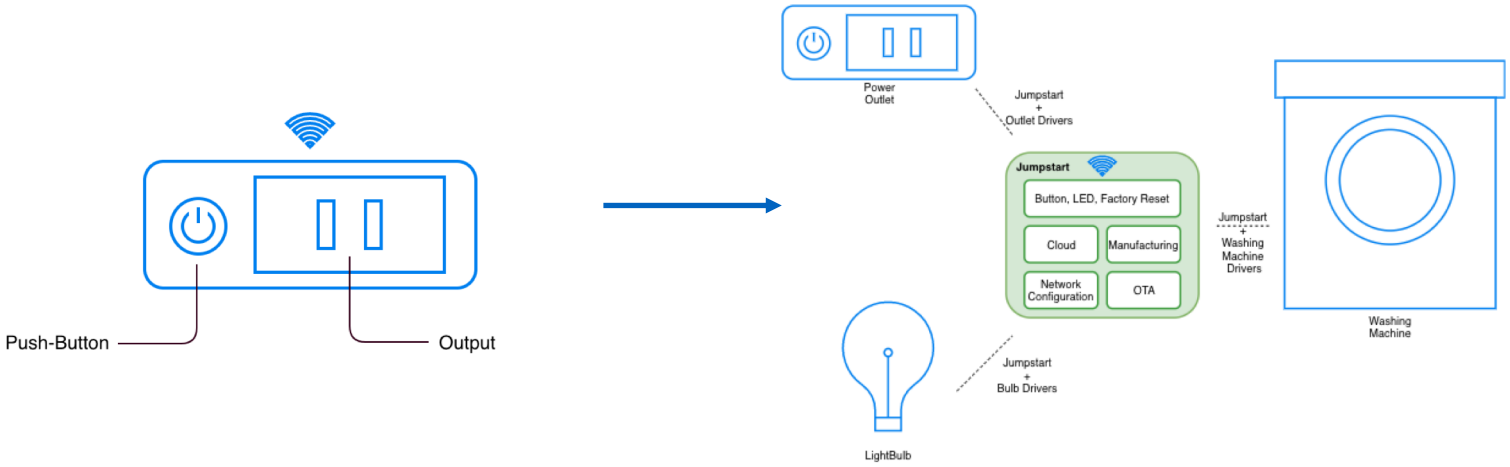
CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Let‘s Start ！

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Getting started

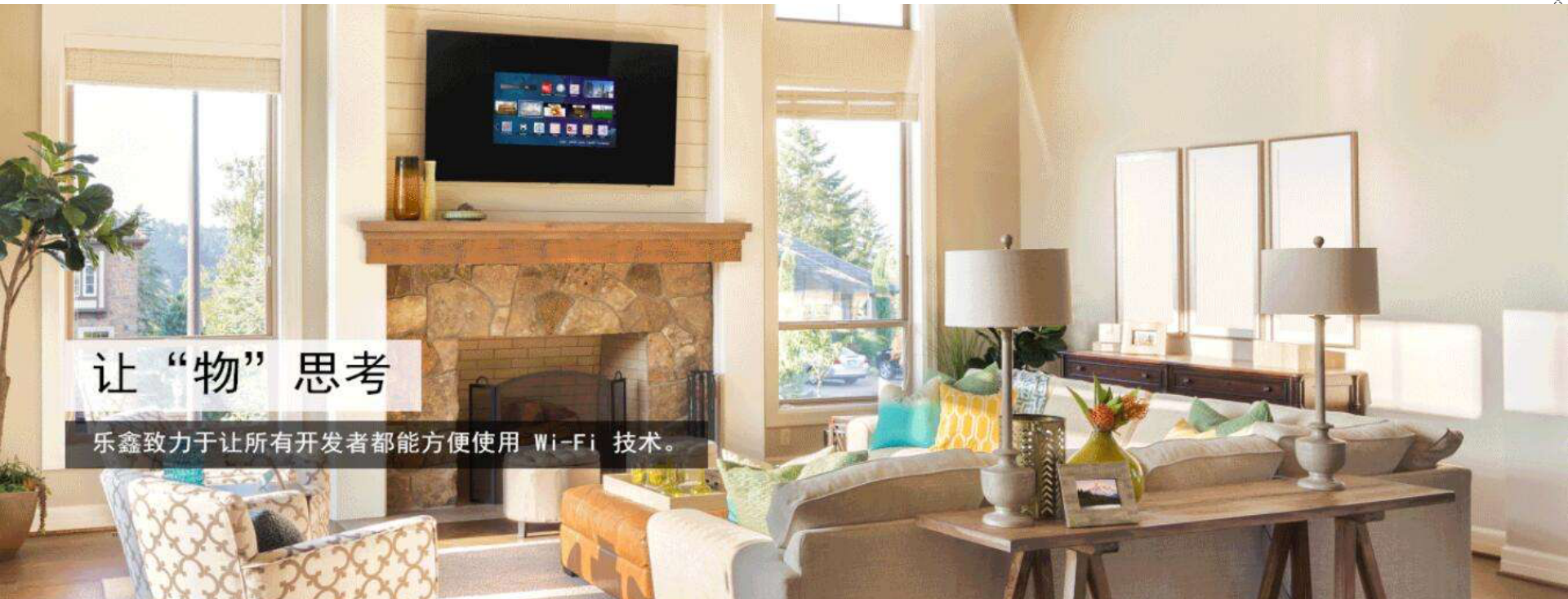
Building Products with ESP32 fast: Jumpstart from concept to production



CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

ESP JumpStart

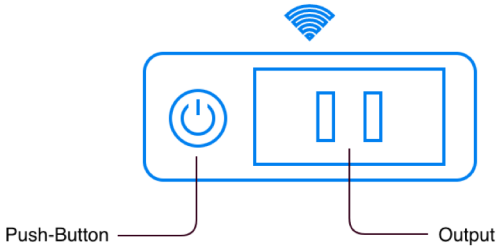
Replace and implement your smart home！



CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

ESP Jumpstart: Objective

- Build a complete “product”: Smart Power Outlet
  - Control GPIO output
  - Network Configuration
  - Remote control through cloud
  - OTA Upgrades
  - Reset to Factory



CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Part 1: Hello World

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

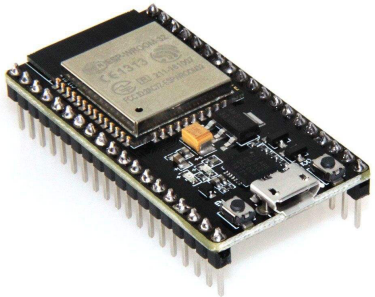
Part 1 Objective

- Preparations
- Development Host Setup
- ESP-IDF
- Build, Flash & Monitor

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Preparations

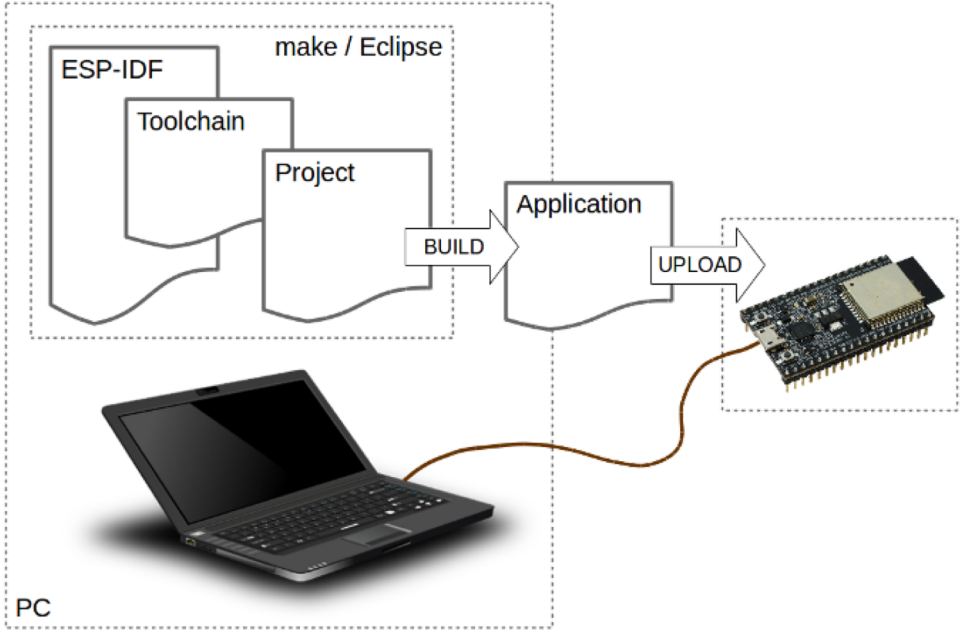
- Development Host PC
  - Windows
  - Linux
  - Mac
- ESP32-DevKitC
- USB to mini-USB cable
- Wi-Fi Access Point



CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Typical Development Setup

The diagram depicts the typical setup for development with ESP32.

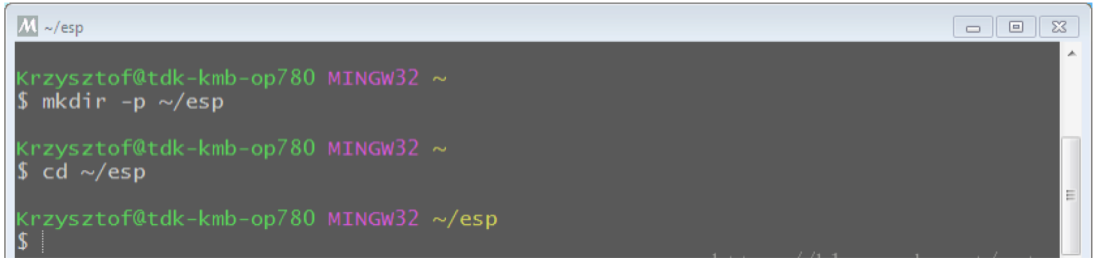


CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

ESP-IDF

ESP-IDF is Espressif's IoT Development Framework for ESP32

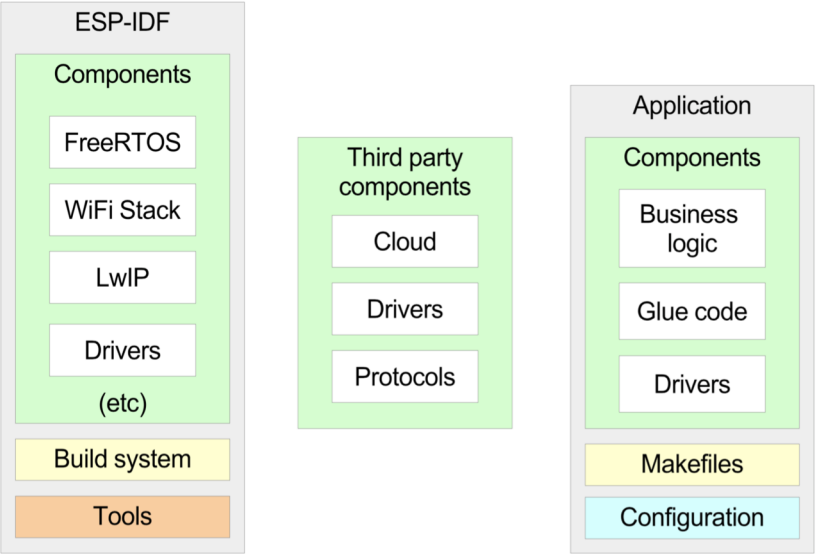
- A collection of Libraries and Header files
- Tools and utilities: build, flash, debug



CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

IDF Details - a component based design

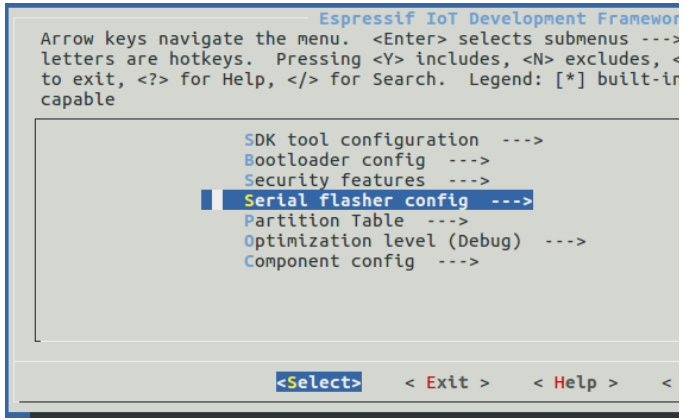
- All the software in the IDF is available as components.



CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

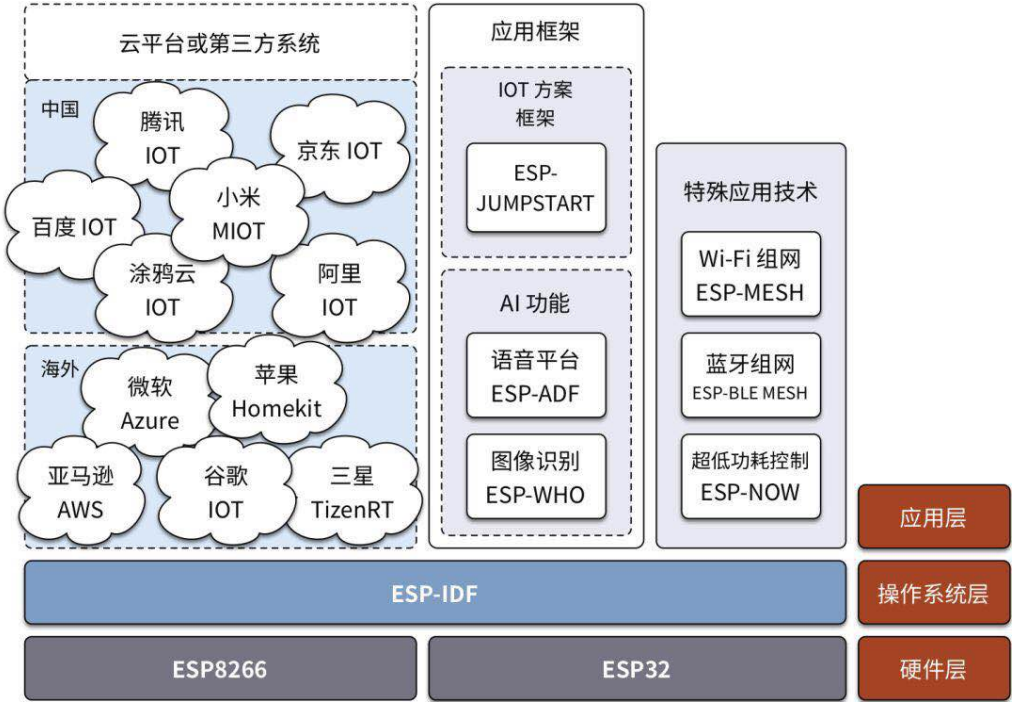
Development Framework Configuration

- Configuration system based on Linux kernel's kconfig
- Configure settings for your project (impacts feature, performance, memory)
- Each component can define its own configuration options
- Use *make menuconfig* command to edit the configuration



CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

基于ESP-IDF的应用框架



CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

应用实例

MESH:

Espressif Wi-Fi Mesh 组网方案

使用乐鑫 Wi-Fi Mesh 组网的智能灯与人脸识别技术结合的应用 Demo

BLE MESH

当 ESP-MESH 应用于节能照明

HMI:

Thermostat for air-conditioner

Control Panel for coffee machine

Audio Player Device

LittlevGL-demo-video

ESP32:

让智能陪伴成长：ESP32 绘本故事机方案

ESP32 生态应用概览(2019.01-02)

ESP32 可移动网络摄像头

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Resources

ESP-IDF Programming Guide

Espressif Systems GitHub

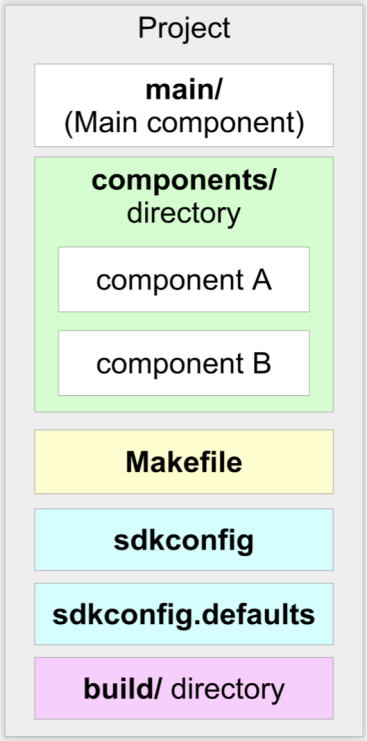
ESP32 Forum

Espressif document

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

IDF Details - Application's Structure

- An application must contain one *main* component.



CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Build, Flash & Monitor

- make flash
- make monitor

```
$ make monitor
MONITOR
--- idf_monitor on /dev/ttyUSB0 115200 ---
--- Quit:Ctrl+] | Menu:Ctrl+T | Help:Ctrl+T followed by Ctrl+H ---
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
ets Jun  8 2016 00:22:57
...
```

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Code

```
#include <stdio.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"


void app_main()
{
    int i = 0;
    while (1) {
        printf("[%d] Hello world!\n", i);
        i++;
        vTaskDelay(5000 / portTICK_PERIOD_MS);
    }
}
```

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Code Explanations

- app_main: Application Entry Point
- Runs in the application thread
- C Library
- FreeRTOS: Real-Time Operating System

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Code Explanation: C standard Library

ESP-IDF uses newlib 2.2.0 as C standard library

- Most of the C standard library is supported, plus extras

- stdlib, stdio, string, math, time
- File/Directory operations
- Reentrant functions
- Timezones, DST

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Code Explanation: FreeRTOS

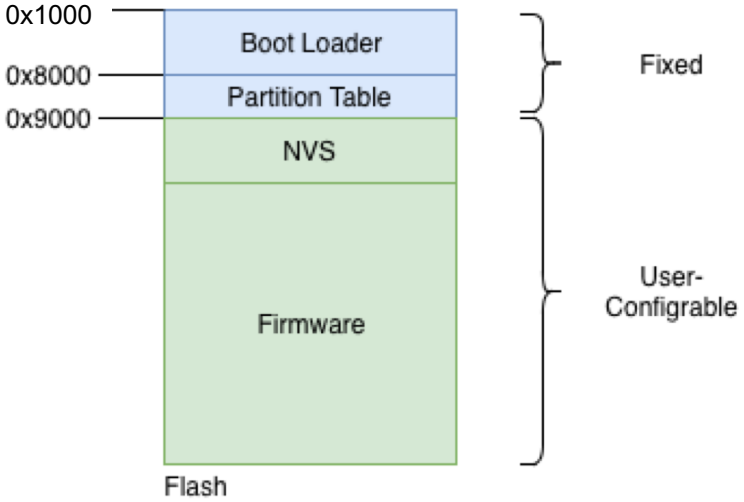
Real-Time Operating System https://www.freertos.org

- Task management
- Message Queues, Semaphores and Mutexes
- Timers

API Reference: https://www.freertos.org/a00106.html
FreeRTOS基础篇 FreeRTOS高级篇

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

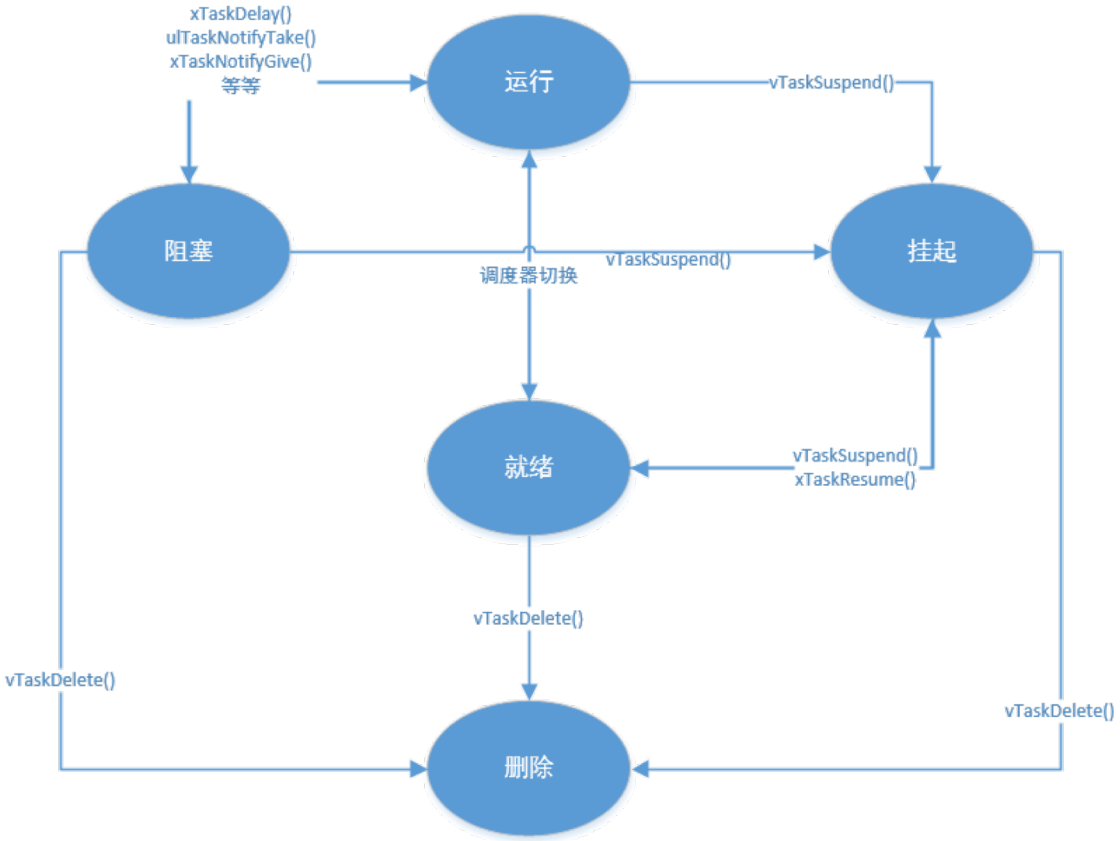
General Notes - Flash



1. First-stage bootloader in ROM loads second-stage bootloader image to RAM (IRAM & DRAM) from flash offset 0x1000.
2. Second-stage bootloader loads partition table and main app image from flash. Main app incorporates both RAM segments and read-only segments mapped via flash cache.
3. Main app image executes. At this point the second CPU and RTOS scheduler can be started.

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

General Notes: Task



CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Part 2: Driver

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Part 2 Objective

Now we will configure a GPIO to act as the output of the power outlet. We will assert this GPIO on or off which would ideally trigger a relay to switch the output on or off.

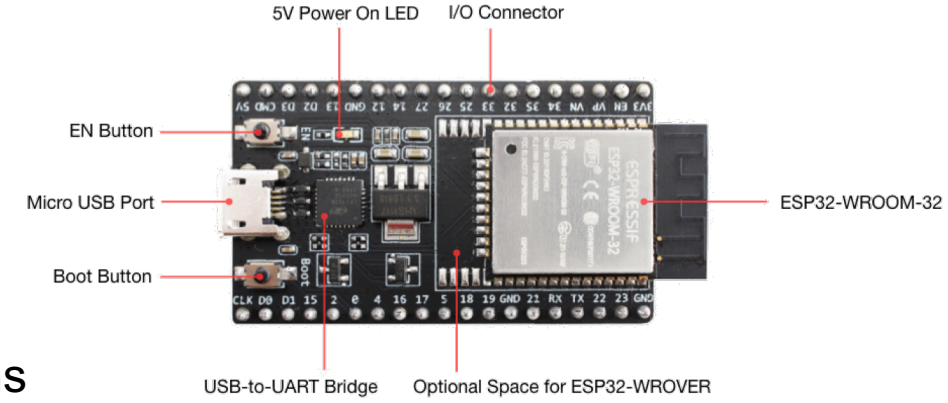
- Configure a push-button
- Use GPIOs
- Configure an output

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Push-Button

Let’s first create a push-button.

- Assert a GPIO
- Callback handler
- iot_button component: Allows long-hold actions



CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

GPIO & Output

- Configure mode
- Set output

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Code

The code for enabling this is shown as below:

```
#include <iot_button.h>


button_handle_t
btn_handle=iot_button_create(BUTTON_GPIO,
                                BUTTON_ACTIVE_LEVEL);


iot_button_set_evt_cb(btn_handle, BUTTON_CB_RELEASE,
                            btn_cb, "RELEASE");
```

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Code

```
gpio_config_t io_conf;
io_conf.mode = GPIO_MODE_OUTPUT;
io_conf.pull_up_en = 1;
io_conf.pin_bit_mask = ((uint64_t)1 << OUTPUT_GPIO);

/* Configure the GPIO */
gpio_config(&io_conf);

/* Assert GPIO */
gpio_set_level(OUTPUT_GPIO, target);
```

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

ESP32 Functional Block Diagram

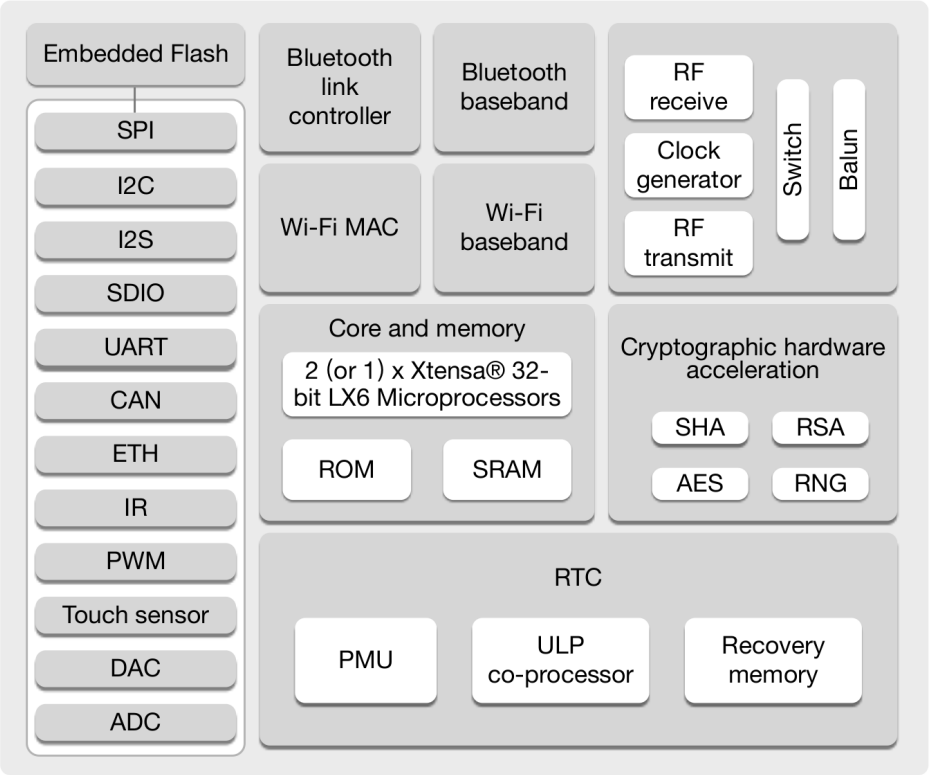


图 1: 功能框图

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

ESP32 Peripherals

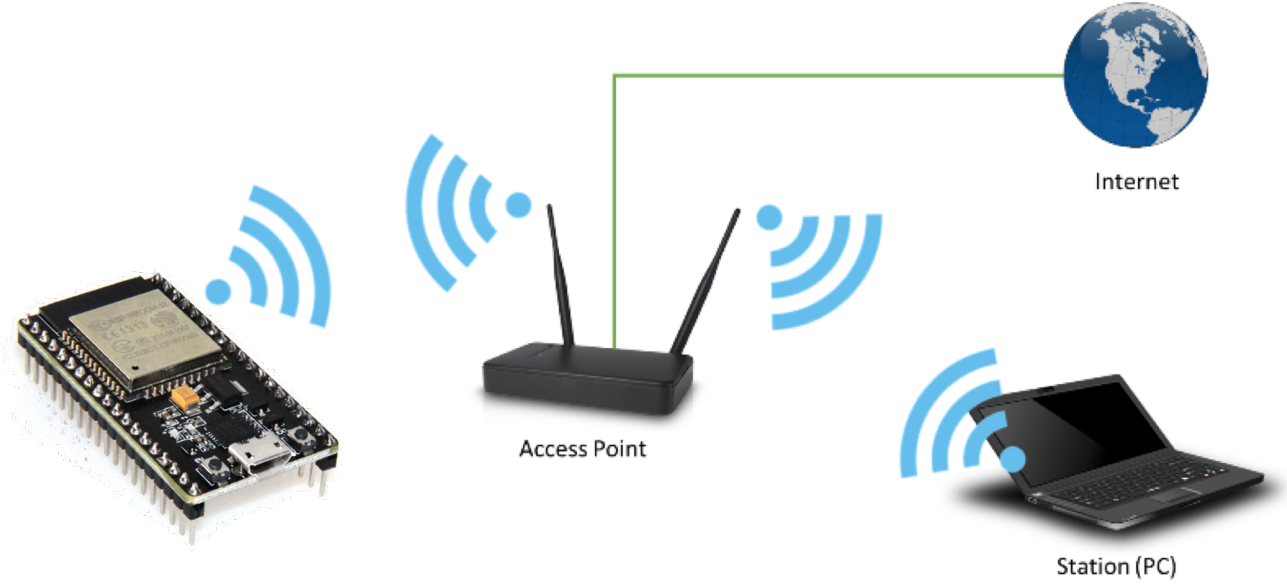
PWM	**16**	**light,motor**
I2C	2 * HW I2C	Sensor
I2S	2	Audio,LCD,Camera
SPI	2	LCD
Timer	4	
RMT	8	
PCNT	8	
SD/MMC	1	SD-Card
UART	3	

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Part 3: Wi-Fi

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

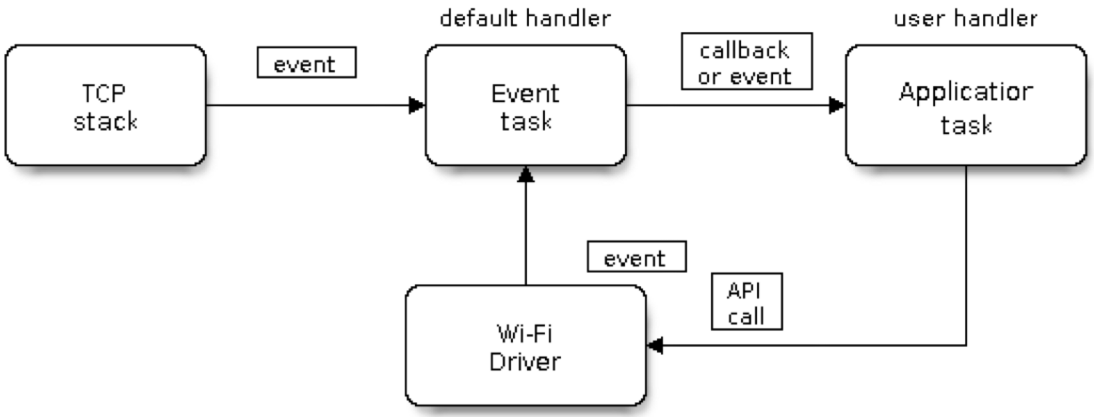
Wi-Fi Introduction



ESP32 operating in the **Station** mode

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

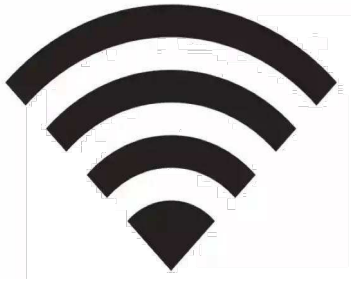
ESP32 Wi-Fi Programming Model



CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Objective

- Configure Wi-Fi station interface
- Connect to pre-configured AP
- Asynchronous event handler:
  - Connected
  - Disconnected
  - IP Address assignment



CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Code

```
#include <esp_wifi.h>
#include <esp_event_loop.h>

tcpip_adapter_init();
esp_event_loop_init(event_handler, NULL);


wifi_init_config_t cfg = WIFI_INIT_CONFIG_DEFAULT();
esp_wifi_init(&cfg);
esp_wifi_set_mode(WIFI_MODE_STA);
```

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Code

```
wifi_config_t wifi_config = {
    .sta = {
        .ssid = EXAMPLE_ESP_WIFI_SSID,
        .password = EXAMPLE_ESP_WIFI_PASS,
    },
 };
esp_wifi_set_config(ESP_IF_WIFI_STA, &wifi_config);

esp_wifi_start();
```

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Code

```
esp_err_t event_handler(void *ctx, system_event_t
*event)
{
    switch(event->event_id) {
    case SYSTEM_EVENT_STA_START:
        esp_wifi_connect();
        break;
    case SYSTEM_EVENT_STA_GOT_IP:
        ESP_LOGI(TAG, "Connected with IP Address:%s",
ip4addr_ntoa(&event->event_info.got_ip.ip_info.ip));
        break;
    case SYSTEM_EVENT_STA_DISCONNECTED:
        esp_wifi_connect();
        break;
    return ESP_OK;
}
```

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Part 4: Network Configuration

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

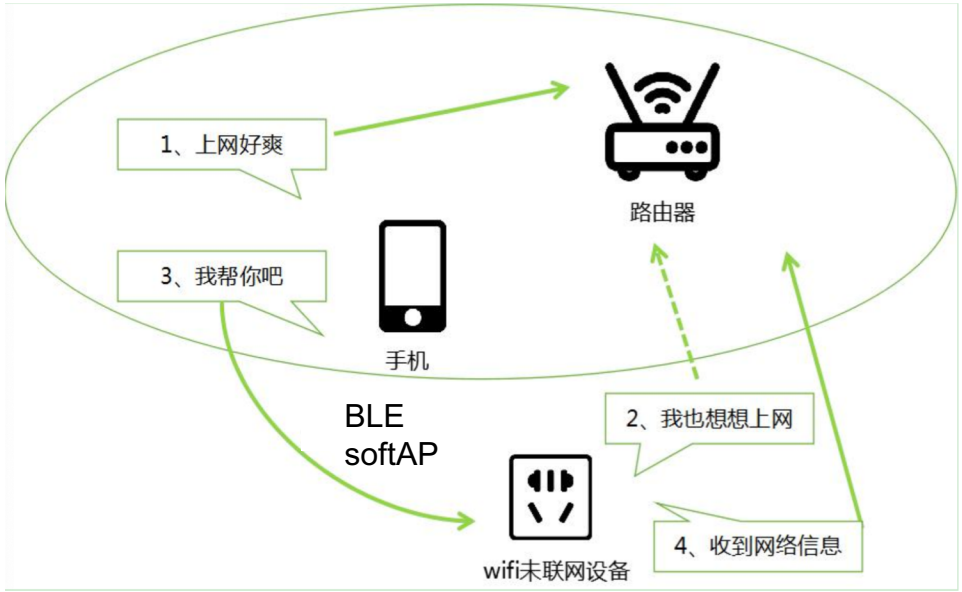
Objective

- Network Provisioning
- NVS: Persistent key-value store
- Reset to Factory

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Provisioning Methods Overview

In this step we will build a firmware such that the end-user can configure his/her Wi-Fi network's credentials into the device at run-time.

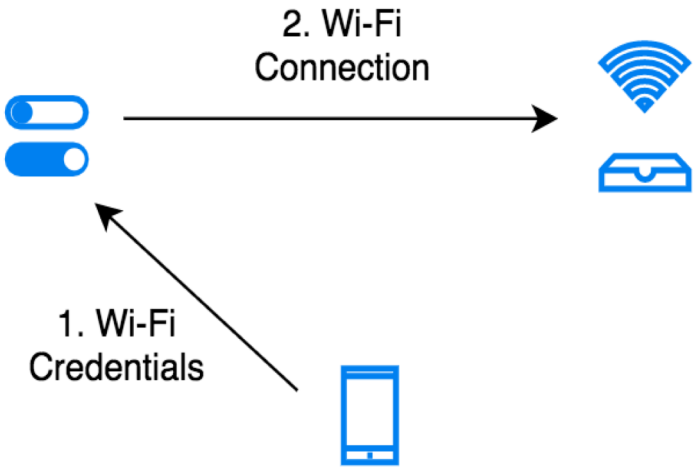


CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Provisioning Methods Overview

The end-user typically uses his/her smart-phone to *securely* configure the Wi-Fi credentials into their device. Once the devices acquire these credentials, it can then connect to his/her home Wi-Fi network.

- Soft AP
- Bluetooth
- Smart-Config



CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Comparison

	Advantages	Disadvantages
SoftAP	1. 非常可靠（SoftAP 只作为 Wi-Fi 接入点，技术成熟） 2. 简洁（无需在设备固件中增加额外代码）。	1. 要求客户手动切换手机连接的 Wi-Fi 网络，用户体验不友好。 2. 通常很难直接将 “切换网络” 的过程写入代码，交由 app 自动完成（全部 iOS 和部分 Android 版本均并不允许手机 app 进行上述操作）。
BLE	1. 用户无需切换 Wi-Fi 2. iOS 和 Android 系统都支持手机 app 扫描并连接到周围的 BLE 设备。	1. 需要加入蓝牙相关代码。这就意味着固件大小会增加，因而会对 flash 提出更高要求。 2. BLE 在配网结束前还会一直占用内存。

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

配网底层结构组件

- 统一配置规范
- IDF 组件
- Android/iOS 手机库
- Reference applications on the device and Android/iOS

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Code

```
if (conn_mgr_prov_is_provisioned(&provisioned) !=
ESP_OK) {
    return;
}

if (! provisioned) {
    /* Starting provisioning */
    conn_mgr_prov_start_provisioning(prov_type,
                security, pop, service_name,
service_key);
} else {
    /* Start the station */
    wifi_init_sta();
}
```

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Code

```
esp_err_t event_handler(void *ctx, system_event_t
*event)
{
     conn_mgr_prov_event_handler(ctx, event);

     switch(event->event_id) {
     case SYSTEM_EVENT_STA_START:
…
…
…
```

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Flash Partitions

- Store user’s configuration: NVS
- Custom Partition Table: partitions.csv
- Update in ‘menuconfig’

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

NVS

The NVS is a software component that maintains a persistent storage of key-value pairs.

- 永久储存键值对 (key-value pairs)
- 可以防止设备断电带来的数据损坏 (metadata corruption)
- 可以通过将写入的内容分布到整个 NVS 分中以处理 flash 磨损的问题。
- Support namespaces
- Used for: manufacturing settings, user’s configuration, state maintenance after reset

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Reset to Factory

Once the user configuration is stored into the NVS as discussed above, **Reset to Factory** behaviour can be achieved by simply erasing the NVS partition.

- Long-press Push Button
- Erase NVS

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Code

```
static void button_press_3sec_cb(void *arg)
{
    nvs_flash_erase();
    esp_restart();
}

iot_button_add_on_press_cb(btn_handle, 3,
                           button_press_3sec_cb, NULL);
```

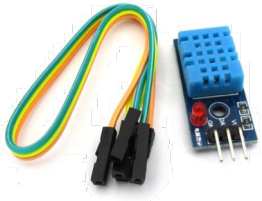
CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Hands-on session & Demo Show

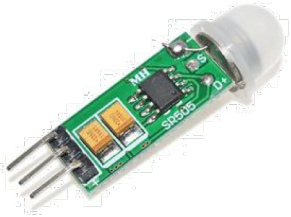
CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Hands-on

- Work in groups of 2 or 3
- Use modules we provide or other modules you like



DHT 11 Humidity sensor
Humidity Sensor: state to PC/smart phone

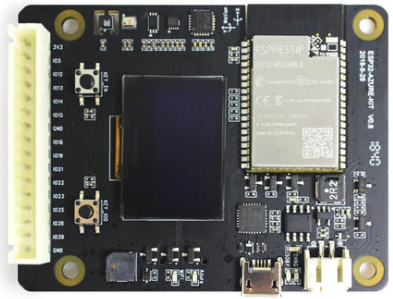


HC-SR505 Mini human body sensor
Human detection: state feedback to PC/smart phone

- It can be a demo, a prototype, even an idea!
- Group Demo Show next week

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

ESP32-Azure IoT Kit

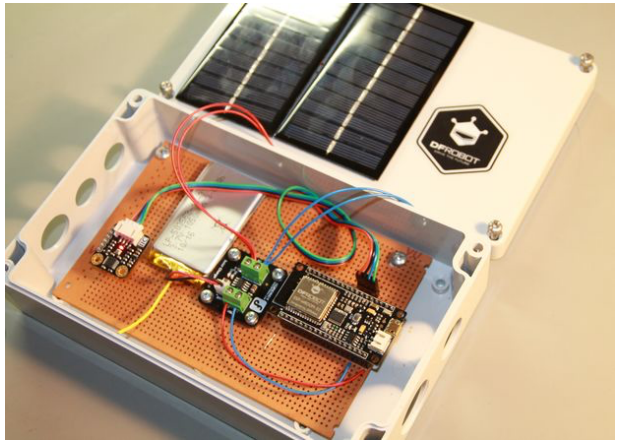


ESP32-Azure IoT Kit

An Espressif development board with powerful functions, designed for sensor data acquisition, cloud platform access and a wide range of applications.

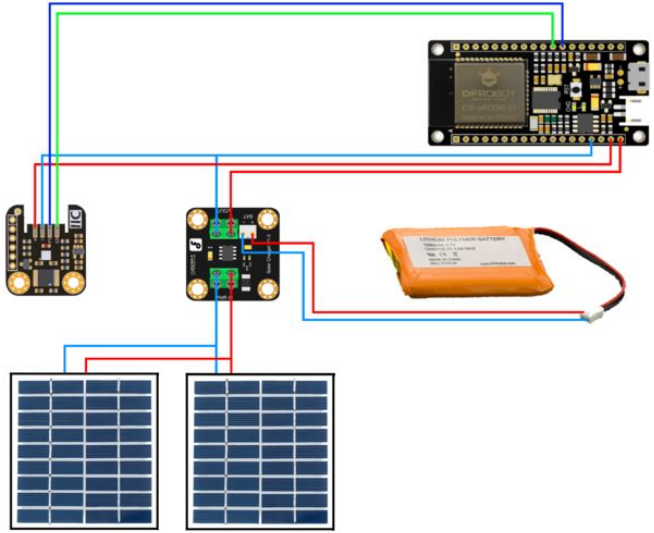
CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Projects from foreign developers



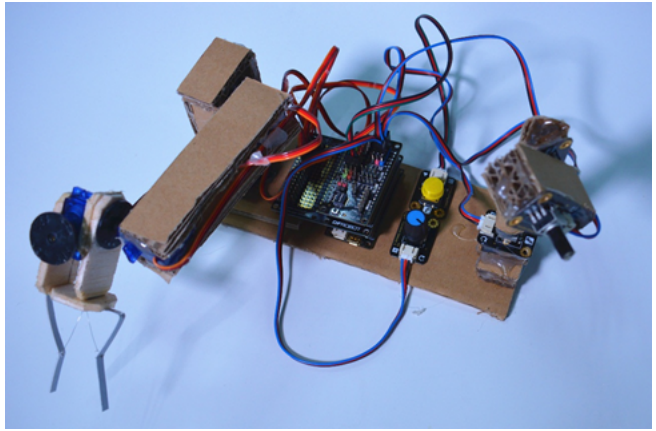
ESP32 Solar Weather Station

https://www.dfrobot.com/blog-699.html



CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Projects from foreign developers



DIY an intelligent robot arm
https://www.dfrobot.com/blog-699.html

A Step-Counter that shows the time
https://www.dfrobot.com/blog-699.html

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL

Demo Show

What’s your idea?

CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL CONFIDENTIAL