

ESP32

网络服务的开发

是馄酱呀 2024年1月

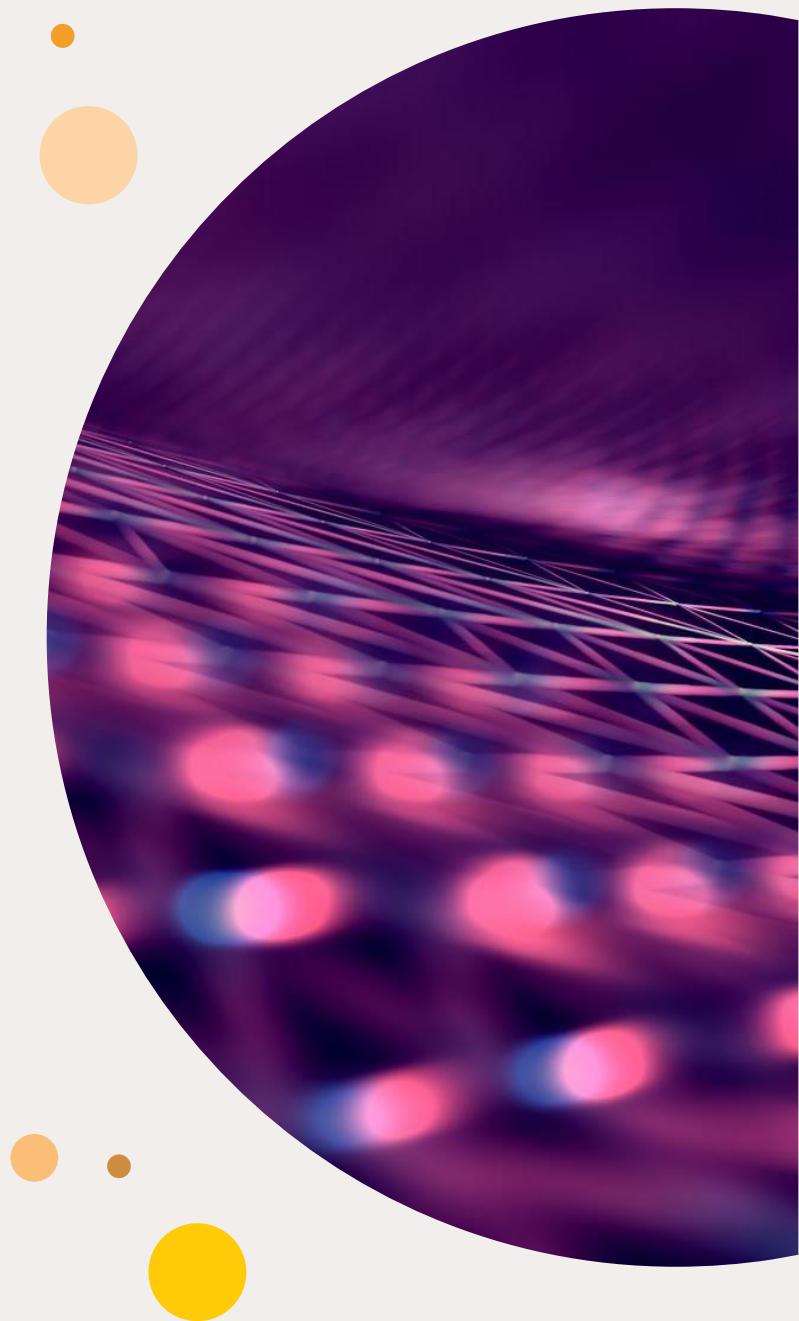


ESP32 WEB系列教程

01·PLATFORMIO配置

ESP32S3分区表与SPIFFS文件系统

是馄酱呀 2024年1月



一. ESP32S3的ROM, RAM 和 FLASH

芯片手册摘录:

CPU 和片上存储器

- 内置 ESP32-S3 系列芯片, Xtensa® 双核 32 位 LX7 微处理器 (支持单精度浮点运算单元), 支持高达 240 MHz 的时钟频率
- 384 KB ROM
- 512 KB SRAM
- 16 KB RTC SRAM
- 最大 16 MB PSRAM

模组集成元件

- 40 MHz 集成晶振
- 最大 16 MB Quad SPI flash

➤ Internal ROM (384 KB)

Internal ROM 是只读存储器，不可编程。 Internal ROM 中存放有一些系统底层软件的 ROM 代码（程序指令和一些只读数据）

➤ Internal SRAM (512 KB)

内部静态存储器（SRAM）是易失性（volatile）存储器，可以快速响应 CPU 的访问请求（通常一个 CPU 时钟周期）

➤ RTC Memory (16 KB)

RTC 存储器以静态 RAM (SRAM) 方式实现，因此也是易失性存储器。但是，在 deepsleep 模式下，存放在 RTC 存储器中的数据不会丢失。

➤ 片外PSRAM和片外Flash

- ESP32-S3 支持以 SPI、Dual SPI、Quad SPI、Octal SPI、QPI、OPI 等接口形式**连接 flash 和片外 RAM**
- **片外Flash可读可写**

-
- ESP32S3芯片手册: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1u_wroom-1u_datasheet_cn.pdf
 - ESP32S3存储器类型: https://docs.espressif.com/projects/esp-idf/zh_CN/latest/esp32s3/api-guides/memory-types.html
 - ESP32S3技术参考手册: https://www.espressif.com/sites/default/files/documentation/esp32-s3_technical_reference_manual_cn.pdf

二. platformio.ini配置文件

➤ PlatformIO是什么

用户友好且可扩展的集成开发环境，配有一组专业开发工具，提供现代且强大的功能，以加速并简化嵌入式产品的创建和交付。

ESP32S3N16R8



-
- PlatformIO官网: <https://platformio.org/>
 - PlatformIO开发环境配置: <https://www.bilibili.com/video/BV1Qd4y1p7EQ>

➤ 开发PlatformIO工程需要的插件



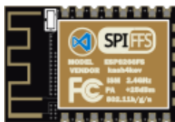
PlatformIO IDE v3.3.3

PlatformIO platformio.org | 4,049,975 | ★★★★★ (3462)

Your Gateway to Embedded Software Development Excellence: CMSIS, ESP

禁用 卸载 设置

此扩展已全局启用。



ESP8266FS v1.1.0

kash4kev | 52,527 | ★★★★★ (2)

Visual Studio Code extension for ESP8266/ESP32 File System (SPIFFS)

禁用 卸载 设置

此扩展已全局启用。



C/C++ v1.18.5

Microsoft microsoft.com | 59,904,506 | ★★★★★ (551)

C/C++ IntelliSense, debugging, and code browsing.

禁用 卸载 切换到预发布版本 设置

此扩展已全局启用。



HTML CSS Support v2.0.9

ecmel | 18,692,440 | ★★★★★ (101) | ❤️ 发起人

CSS IntelliSense for HTML

禁用 卸载 设置

此扩展已全局启用。



Auto Rename Tag v0.1.10

Jun Han | 16,849,268 | ★★★★★ (194)

Auto rename paired HTML/XML tag

禁用 卸载 设置

此扩展已全局启用。



Live Server v5.7.9

Ritwick Dey | 44,852,002 | ★★★★★ (473)

Launch a development local Server with live reload feature for static

禁用 卸载 设置

此扩展已全局启用。

➤ platform.ini配置文件

```
1  [env:esp32-s3-devkitc-1]
2  ; 设置板子编译链
3  platform          = espressif32 @ ^6.5.0
4  board             = esp32-s3-devkitc-1
5  framework         = arduino
6  platform_packages = espressif/toolchain-xtensa-esp32s3@8.4.0+2021r2-patch5
7  build_unflags     = -std=gnu++20
8  ; 启用 8MB 的 PSRAM
9  build_flags       = -DBOARD_HAS_PSRAM
10 board_build.arduino.memory_type = qio_opi
11 ; 启用 16MB 的 ROM (FLASH)
12 board_build.arduino.partitions = default_16MB.csv ; 指定为16MB的FLASH分区表
13 board_upload.flash_size        = 16MB             ; 指定FLASH容量为16MB
14 ; 启用 spiffs 文件系统
15 board_build.filesystem = spiffs
16 extra_scripts          = replace_fs.py
17 ; 调试相关 (启用JTAG调试)
18 upload_speed           = 2000000
19 upload_port            = COM5                 ; USB串行设备 (JTAG) 的串口号
20 debug_tool             = esp-builtin
21 debug_init_break       = break setup
22 monitor_port           = COM7                 ; CH340系列的串口号
23 monitor_speed          = 115200
24 ; 依赖库
25 lib_deps =
26 |     esphome/ESPAsyncWebServer-esphome@^3.1.0
27 |     esphome/AsyncTCP-esphome@^2.1.2
```

➤ platform.ini配置文件解读

Step1: 编译链的配置

```
1 [env:esp32-s3-devkitc-1]
2 ; 设置板子编译链
3 platform          = espressif32 @ ^6.5.0
4 board             = esp32-s3-devkitc-1
5 framework         = arduino
6 platform_packages = espressif/toolchain-xtensa-esp32s3@8.4.0+2021r2-patch5
7 build_unflags     = -std=gnu++20
```

platformio/espressif32 by PlatformIO PLATFORM TOP 10 OFFICIAL

6.5.0 • Public • Published 2 months ago • dev-platform wi-fi bluetooth xtensa risc-v

ESP32 is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and Bluetooth. ESP32 integrates an antenna switch, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules.

Readme	Installation	Boards 227	Frameworks 2	Examples 19	Insights
All 50 versions of espressif32 since November 18, 2016.					
Major	Version	Size	Published		
6.x	6.5.0 Latest	1.35 MiB	2 months ago	Release Notes	
6.x	6.4.0	1.35 MiB	5 months ago	Release Notes	
5.x	5.4.0	297 KiB	8 months ago	Release Notes	
6.x	6.3.2	1.35 MiB	8 months ago	Release Notes	
6.x	6.3.1	1.35 MiB	9 months ago	Release Notes	
6.x	6.3.0	1.35 MiB	9 months ago	Release Notes	
6.x	6.2.0	1.35 MiB	10 months ago	Release Notes	

Homepage
espressif.com

Repository
github.com/platformio/platform-espressif32

Popularity
1 1

License
Apache-2.0

Stars
771

Issues
154

Last Modified
2 months ago

Installed Size
6.96 MiB

Versions 50
6.5.0 Latest
2 months ago

espressif/toolchain-xtensa-esp32s3 by Espressif Systems TOOL TOP 100 VERIFIED

12.2.0+20230208 • Public • Published 10 months ago • build tools compiler assembler linker preprocessor + 3 categories

GCC Toolchain for Espressif ESP32-S3 Xtensa MCUs

【注】1. 最新的11和12工具链只针对IDF框架
2. 不能使用espressif/toolchain-xtensa-esp32工具链

Major	Version	Size	Published	20 1	GPL-2.0-or-later
12.x	12.2.0+20230208 Latest	128 MiB	10 months ago	Release Notes	
11.x	11.2.0+2022r1	93.6 MiB	a year ago	Release Notes	
8.x	8.4.0+2021r2-patch5	80.8 MiB	a year ago	Release Notes	
8.x	8.4.0+2021r2-patch3	112 MiB	2 years ago	Release Notes	
8.x	8.4.0+2021r2-patch2	112 MiB	2 years ago	Release Notes	

Stars
93

Issues
11

Last Modified
10 months ago

Installed Size
395 MiB

Versions 5
12.2.0+20230208 Latest
10 months ago

- espressif32: <https://registry.platformio.org/platforms/platformio/espressif32/versions>
- espressif/toolchain-xtensa-esp32s3: <https://registry.platformio.org/tools/espressif/toolchain-xtensa-esp32s3/versions>

Step2: 配置系统调试与下载工具

```
8 ; 调试相关（启用JTAG调试）
9 upload_speed      = 2000000
10 upload_port       = COM5           ; USB串行设备（JTAG）的串口号
11 debug_tool        = esp-builtin
12 debug_init_break  = break setup
13 monitor_port      = COM7           ; CH340系列的串口号
14 monitor_speed     = 115200
```

详细配置过程可以参考另一个UP主的视频



-
- ESP32自带JTAG: 最简Arduino断点单步调试指南: <https://www.bilibili.com/video/BV1eM411d72C>

Step3: 启用片外PSRAM和ROM

```
8 ; 启用 8MB 的 PSRAM
9 build_flags = -DBOARD_HAS_PSRAM
10 board_build.arduino.memory_type = qio_opi
11 ; 启用 16MB 的 ROM (FLASH)
12 board_build.arduino.partitions = default_16MB.csv ; 指定为16MB的FLASH分区表
13 board_upload.flash_size = 16MB ; 指定FLASH容量为16MB
```

ESP32S3 对应关系

订购代码 ²	Flash ^{3, 4}	PSRAM ⁵	环境温度 ⁶ (°C)	模组尺寸 ⁷ (mm)
ESP32-S3-WROOM-1-N4	4 MB (Quad SPI)	-	-40 ~ 85	18.0 × 25.5 × 3.1
ESP32-S3-WROOM-1-N8	8 MB (Quad SPI)	-	-40 ~ 85	
ESP32-S3-WROOM-1-N16	16 MB (Quad SPI)	-	-40 ~ 85	
ESP32-S3-WROOM-1-H4	4 MB (Quad SPI)	-	-40 ~ 105	
ESP32-S3-WROOM-1-N4R2	4 MB (Quad SPI)	2 MB (Quad SPI)	-40 ~ 85	
ESP32-S3-WROOM-1-N8R2	8 MB (Quad SPI)	2 MB (Quad SPI)	-40 ~ 85	
ESP32-S3-WROOM-1-N16R2	16 MB (Quad SPI)	2 MB (Quad SPI)	-40 ~ 85	
ESP32-S3-WROOM-1-N4R8	4 MB (Quad SPI)	8 MB (Octal SPI)	-40 ~ 65	
ESP32-S3-WROOM-1-N8R8	8 MB (Quad SPI)	8 MB (Octal SPI)	-40 ~ 65	
ESP32-S3-WROOM-1-N16R8	16 MB (Quad SPI)	8 MB (Octal SPI)	-40 ~ 65	
ESP32-S3-WROOM-1-N16R16V ⁸	16 MB (Quad SPI)	16 MB (Octal SPI)	-40 ~ 65	

- ESP32S3芯片手册: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_cn.pdf
- Code+platformio配置ESP32-S3-N16R8（8MB PSRAM + 16MB FLASH）工程: <https://www.cnblogs.com/macrored/p/17357581.html>

分区表与Flash分区（1）：默认分区

default_16MB.csv 文件

1	#	Name,	Type,	SubType,	Offset,	Size,	Flags
2	nvs,	data,	nvs,	0x9000,	0x5000,		
3	otadata,	data,	ota,	0xe000,	0x2000,		
4	app0,	app,	ota_0,	0x10000,	0x640000,		
5	app1,	app,	ota_1,	0x650000,	0x640000,		
6	spiffs,	data,	spiffs,	0xc90000,	0x360000,		
7	coredump,	data,	coredump,	0xFF0000,	0x10000,		



起始地址（Offset）

0x000000	
0x001000	
0x009000	BootLoader (28KB)
0x00E000	nvs (20KB)
0x010000	otadata (8KB)
0x065000	app0 (6.25MB)
0x0C9000	app1 (6.25MB)
0xFF0000	spiffs (3.375MB)
	coredump (64KB)

- Size大小的换算举例： $0x640000 \text{ Byte} = 6553600 \text{ Byte} = \frac{6553600}{1024 \times 1024} \text{ MB} = 6.25 \text{ MB}$
- 0x001000之前不存放数据
- **nvs**: 存放WiFi相关的数据
- **otadata**: 系统从哪个app分区启动由此分区内的信息决定
- **app0**和**app1**分区: 用于保存OTA（空中升级）下载的固件。OTA启用后，OTA下载的固件镜像交替保存于OTA_0/OTA_1分区，分区，镜像验证后，OTA data分区更新，指定在下一次启动时使用该镜像。OTA不会影响到Factory app分区（后面介绍），这样用户可以随时恢复到出厂状态。
- **spiffs**分区用于存放spiffs文件系统
- **core dump**分区用于查找系统崩溃时的软件错误，系统崩溃的时候会将调试信息写入到Flash中保存以便对崩溃原因进行分析。

分区表与Flash分区（2）：其他分区类型与factory分区

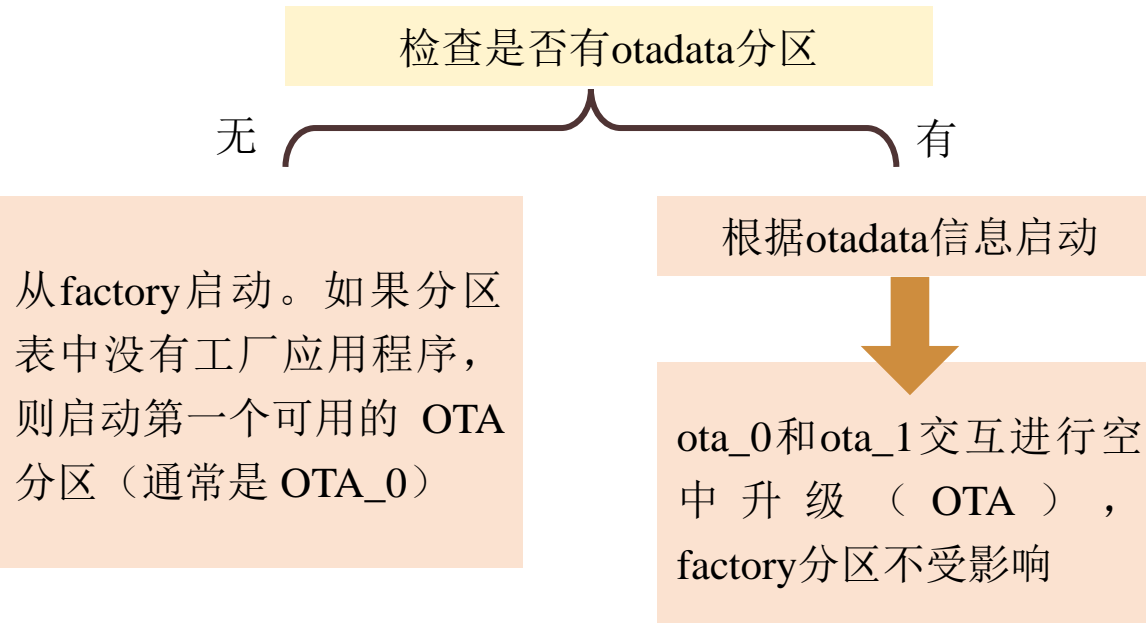
Single factory app, no OTA

```
# ESP-IDF Partition Table
# Name,   Type, SubType, Offset,   Size,   Flags
nvs,      data, nvs,      0x9000, 0x6000,
phy_init, data, phy,      0xf000, 0x1000,
factory,  app,  factory, 0x10000, 1M,
```

- factory用于保存工厂（出厂）应用程序。
- ESP32的启动流程如下

Factory app, two OTA definitions

```
# ESP-IDF Partition Table
# Name,   Type, SubType, Offset,   Size,   Flags
nvs,      data, nvs,      0x9000, 0x4000,
otadata,  data, ota,      0xd000, 0x2000,
phy_init, data, phy,      0xf000, 0x1000,
factory,  app,  factory, 0x10000, 1M,
ota_0,    app,  ota_0,    0x110000, 1M,
ota_1,    app,  ota_1,    0x210000, 1M,
```



- ESP32分区表分析总结: <https://goodmemory.cc/esp32-partition-table/>
- 分区表-ESP-IDF编程指南: https://docs.espressif.com/projects/esp-idf/zh_CN/latest/esp32s3/api-guides/partition-tables.html
- ESP32分区表图解: <https://blog.csdn.net/toopoo/article/details/107327828>
- ESP32的调试工具coredump: <https://blog.csdn.net/tidyjiang/article/details/72123346>

Step4: 启用spiffs文件系统

```
21 ; 启用 spiffs 文件系统
22 board_build.filesystem = spiffs
23 extra_scripts           = replace_fs.py
```

Windows默认文件系统：NTFS



SPIFFS文件系统

- 在指定的地址保存数据而不会因为重新更新而删除（如校准数据，步态数据）
- 不用外置的SD卡，节约了硬件资源
- 保存HTML和CSS文件以建立Web服务器
- 保存图像，音频等文件
- 目前，SPIFFS 尚不支持目录，但可以生成扁平结构。如果 SPIFFS 挂载在 /spiffs 下，在 /spiffs/tmp/myfile.txt 路径下创建一个文件则会在 SPIFFS 中生成一个名为 /tmp/myfile.txt 的文件，而不是在 /spiffs/tmp 下生成名为 myfile.txt 的文件；

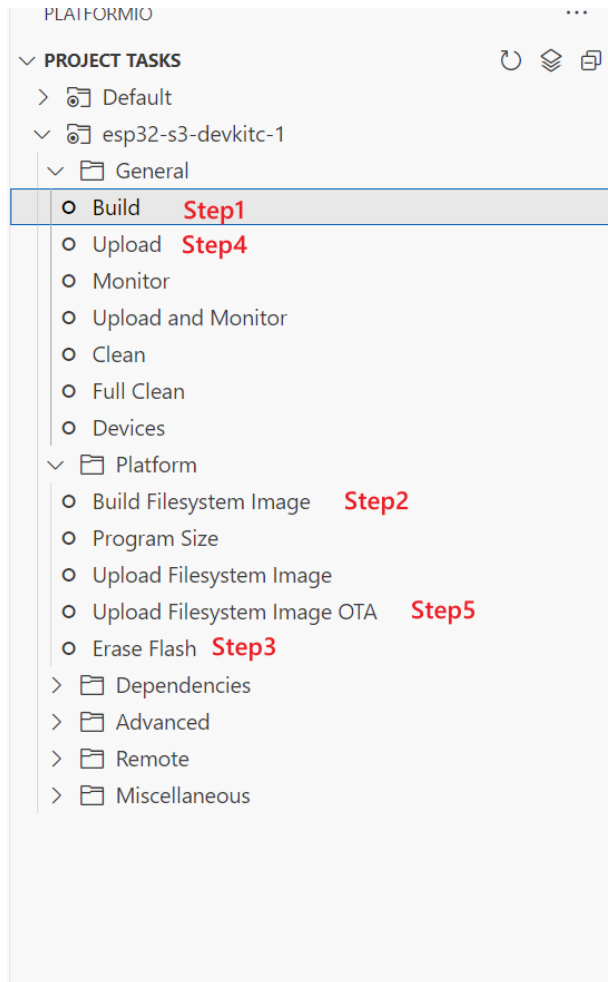
-
- 文件系统SPIFFS: <https://docs.petoi.com/v/chinese/li-cheng/12.-wen-jian-xi-tong-spiffs>
 - Vscod.platformIO.spiffs基础配置: https://blog.csdn.net/weixin_43114955/article/details/106158294
 - 三分钟上马 ESP32 spiffs文件系统: https://blog.csdn.net/weixin_44821644/article/details/109480902
 - 使用Arduino开发ESP32（12）：文件和文件系统使用（基于SPIFFS）: https://blog.csdn.net/Naisu_kun/article/details/90442925

实验1_1：检测片外PSRAM和ROM启动成功

源码：

```
1  #include <Arduino.h>
2  #include <SPIFFS.h>
3  #include "log.h"
4
5  void setup()
6  {
7      Serial.begin(115200);
8      Serial.println();
9
10     LOG_DEBUG("Default free size in RAM: ");
11     LOGLN(heap_caps_get_free_size(MALLOC_CAP_DEFAULT));
12     LOG_DEBUG("Flash size: ");
13     LOGLN(ESP.getFlashChipSize());
14
15     SPIFFS.begin(true);
16     LOGLN_INFO("Start SPIFFS success!");
17 }
18
19 void loop(){
20     vTaskDelay(1000);
21 }
```

编译步骤：



现象：

```
ESP-ROM:esp32s3-20210327
Build:Mar 27 2021
rst:0x1 (POWERON),boot:0x28 (SPI_FAST_FLASH_BOOT)
SPIWP:0xee
mode:DIO, clock div:1
load:0x3fce3808,len:0x44c
load:0x403c9700,len:0xbd8
load:0x403cc700,len:0x2a80
entry 0x403c98d0

DEBUG: Default free size in RAM: 8757975
DEBUG: Flash size: 16777216
INFO: Start SPIFFS success!
```

- 实验源码：https://github.com/sunshineharry/ESP32_WebServer