ESP32-S3 com PlatformIO

Projeto base e diagnóstico de flash, PSRAM e USB - manual prático



Placa de referência	ESP32-S3 módulo N16R8 (16 MB flash, 8 MB PSRAM octal)
Framework	Arduino Core, via PlatformIO
Ambientes prontos	3, comentados no platformio.ini
Nível	Intermediário
Tutorial online	iiot.esp32.tec.br/tutoriais/esp32-s3-com-platformio-do-zero

Os valores de configuração vem da definição de placa usada nos projetos do AnalictY. O sketch de diagnóstico foi escrito para o tutorial e **não foi validado em bancada**: teste na sua placa antes de assumir os resultados.

1. O que significa N16R8

A sigla depois do nome do módulo descreve a memória, e ela determina metade da sua configuração. O N é a flash: N16 são 16 MB. O R é a PSRAM: R8 são 8 MB. Um módulo sem a parte R simplesmente não tem PSRAM.

O detalhe que causa mais dor de cabeça é invisível na sigla: nas variantes R8 a PSRAM é **octal** (OPI, oito linhas de dados), enquanto nas R2 costuma ser **quad** (QSPI, quatro linhas). Declarar octal como quad faz a placa reiniciar em loop ou ignorar os 8 MB em silencio, sem nenhuma mensagem de erro.

A tabela de decisão

Placa	memory_type	Observação
N16R8, N8R8	qio_opi	PSRAM octal; é o caso deste projeto
N8R2, N4R2	qio_qspi	PSRAM quad
sem PSRAM	qio	remova também a flag BOARD_HAS_PSRAM

2. As três linhas que decidem tudo

memory_type

```
board_build.arduino.memory_type = qio_opi
```

Descreve em uma palavra como flash e PSRAM conversam com o chip: primeiro a flash, depois a PSRAM. É a linha mais importante do arquivo.

A flag BOARD_HAS_PSRAM

Só declarar o memory_type não basta. Sem a flag, a PSRAM é inicializada mas as alocações continuam usando apenas a RAM interna - cerca de 320 KB em vez de 8 MB. É a diferença entre não conseguir montar um buffer de imagem e trabalhar com folga.

> O sketch de diagnóstico detecta e avisa quando a flag está ausente.

As portas USB

O ESP32-S3 tem USB nativo, e por isso a maioria dos kits traz duas portas USB-C: uma marcada UART, que passa por um conversor serial, e outra ligada direto no chip. As duas gravam firmware. Só uma entrega o Serial.

Flag	Serial sai por	Ambiente do projeto
ARDUINO_USB_CDC_ON_BOOT=0	porta marcada UART	esp32-s3-n16r8
ARDUINO_USB_CDC_ON_BOOT=1	porta USB nativa	esp32-s3-n16r8-usb

Com o valor 1 a porta some e reaparece após cada gravação, e as primeiras mensagens se perdem nesse intervalo - dai a espera de 1,5 s no início do sketch.

> **Se o monitor está mudo, troque de porta física antes de mexer no código.** Resolve na maioria dos casos.

3. Passo a passo

Passo 1 - Abrir o projeto

Extraia o pacote e abra a pasta no VS Code com a extensão PlatformIO IDE instalada. Todos os arquivos de configuração já estão prontos e comentados.

Passo 2 - Escolher o ambiente

São três, e a escolha depende de qual porta USB você vai usar e de preferir ou não a definição de placa própria.

```
pio run -e esp32-s3-n16r8 -t upload
pio device monitor -b 115200
```

Passo 3 - Ler o diagnóstico

Numa N16R8 configurada corretamente você deve ver:

```
Tamanho : 16777216 bytes (16 MB)
Detectada : 8388608 bytes (8 MB)
Alocacao 1 MB : OK
CDC_ON_BOOT = 0 : Serial sai pela UART.
```

Flash menor que 16 MB significa flash_size declarado abaixo do físico. PSRAM não detectada significa memory_type errado ou flag ausente. O teste de alocação de 1 MB importa porque reconhecer a PSRAM é uma coisa e conseguir usá-la é outra.

Passo 4 - Aproveitar os 16 MB

A tabela padrão foi desenhada para modulos de 4 MB. Numa N16 ela desperdica mais de dois terços do chip, e o compilador ainda reclama de imagem grande demais enquanto sobram 12 MB parados.

```
nvs, data, nvs, 0x9000, 0x5000,
otadata, data, ota, 0xe000, 0x2000,
app0, app, ota_0, 0x10000, 0x300000,
app1, app, ota_1, , 0x300000,
spiffs, data, spiffs, , 0x900000,
```

Dois espaços de aplicação de 3 MB são o requisito para atualização OTA, e sobram 9 MB para SPIFFS. Se você não pretende usar OTA, um único espaço de 6 MB também é válido - o importante é não deixar a tabela de 4 MB numa placa de 16.

Passo 5 - Registrar a placa, se for reutilizar

Repetir dez linhas de board_build em cada projeto novo cansa e convida a erro. O terceiro ambiente mostra a alternativa: registrar a definição uma vez em boards/ e reduzir o platformio.ini a duas linhas.

O PlatformIO procura nessa pasta antes das placas oficiais, então a definição viaja junto com o repositório e o próximo desenvolvedor compila sem configurar nada.

4. Solução de problemas

Sintoma	Causa provável	O que fazer
Reinicia em loop após gravar	memory_type incompatível	confirme se o módulo é R8 (octal, qio_opi) ou R2 (quad, qio_qspi)
Monitor serial mudo	porta USB errada	troque de porta física antes de mexer no código
PSRAM aparece como zero	falta BOARD_HAS_PSRAM	ou o memory_type está errado; rode o diagnóstico
Failed to connect ao gravar	fora do modo de gravação	segure BOOT, toque RESET, solte BOOT e grave de novo
Gravação falha no meio	upload_speed agressivo	921600 é agressivo para cabo longo; baixe para 460800
Sketch too big com 16 MB	tabela de partições de 4 MB	aponte o partitions.csv no platformio.ini
Mudei a configuração e nada muda	pasta build antiga	apague build e recompile

5. Checklist final

	Item
[ ]	memory_type conferido contra a variante física do módulo
[ ]	Flag -DBOARD_HAS_PSRAM presente, se a placa tem PSRAM
[ ]	flash_size declarado igual ao físico
[ ]	partitions.csv apontado no platformio.ini
[ ]	Ambiente escolhido conforme a porta USB em uso
[ ]	Diagnóstico gravado e lido antes de começar o projeto
[ ]	Flash reportou o tamanho esperado
[ ]	PSRAM detectada com o tamanho esperado
[ ]	Teste de alocação de 1 MB passou
[ ]	Serial saindo pela porta que você está usando