Primeiro projeto no PlatformIO

Um arquivo, três placas, nenhum if de hardware - manual prático



Placas	ESP32, ESP32-C3 e ESP32-S3 - o mesmo main.cpp
Framework	Arduino Core, via PlatformIO
Ambientes prontos	4, comentados no platformio.ini
Nível	Iniciante
Artigo online	iiot.esp32.tec.br/tutoriais/o-que-e-platformio

Projeto escrito como demonstração das quatro camadas do PlatformIO. **Não foi validado em bancada** nas três placas: teste na sua antes de assumir os pinos de LED, que variam entre fabricantes mesmo dentro do mesmo modelo de chip.

1. O que o PlatformIO resolve

O platformio.ini é o único lugar onde a configuração do projeto vive. Nada de menu de IDE, nada de **funciona na minha máquina**: quem clonar o repositorio recebe exatamente o mesmo compilador, as mesmas bibliotecas e as mesmas versões que você usou.

As quatro camadas, no arquivo

Linha	Camada	O que ela resolve
platform	plataforma	qual toolchain e SDK baixar
framework	framework	Arduino Core ou ESP-IDF
board	placa	frequência, tamanho de flash, mapa de memória
lib_deps	bibliotecas	quais, e em que versão

O board não é enfeite. É por ele que o PlatformIO sabe que aquele modelo tem tantos MB de flash e tanta RAM, e monta o mapa de memória de acordo.

2. O que este projeto demonstra

Heranca com extends

A seção [comum] guarda o que é igual em todos os ambientes. Cada env herda com extends, então mudar o monitor_speed num lugar muda em todos.

```
build_flags =
${comum.build_flags}
-D LED_PIN=2
```

> Para acrescentar sem substituir, referencie o valor herdado. Sem o ${comum.build_flags} o valor herdado seria **perdido** - é um erro silencioso comum.

Varios ambientes, o mesmo código

Três placas, três env, um main.cpp. O trecho específico de cada uma é ligado por definição de compilador, não por arquivo duplicado nem por linha comentada na mão.

O main.cpp inclusive falha na compilação com mensagem clara se você esquecer de escolher um ambiente. Melhor errar cedo e com explicação do que compilar errado.

Versões travadas

```
bblanchon/ArduinoJson@^7.2.1
```

O acento circunflexo aceita correções e novidades compatíveis, mas nunca uma versão maior que possa quebrar. Sem isso, um projeto que compila hoje pode não compilar em seis meses sem que ninguém tenha mudado uma linha.

Filtros de monitor

```
monitor_filters = esp32_exception_decoder, time
```

O primeiro converte o endereço de um crash na linha exata do seu código - sozinho já paga o PlatformIO. O segundo carimba a hora em cada linha do log.

3. Comandos e estrutura

Comando	O que faz
pio run	compila o ambiente padrão
pio run -e esp32-c3	compila um ambiente específico
pio run -t upload	compila e grava na placa
pio run -t clean	apaga os artefatos de build
pio device list	lista as portas seriais detectadas
pio device monitor	abre o monitor com as opções do arquivo
pio pkg install	baixa as bibliotecas declaradas
pio pkg outdated	mostra dependências com versão nova disponível

Para gravar numa porta específica, acrescente --upload-port COM5 no Windows ou --upload-port /dev/ttyUSB0 no Linux.

Estrutura de pastas

Pasta	Para que serve
src/	seu código; é o que o PlatformIO compila
lib/	bibliotecas suas, uma por subpasta, compiladas sozinhas
include/	cabecalhos compartilhados; crie se precisar
test/	testes unitarios; crie se precisar
.pio/	artefatos de build; NÃO versione

> Acrescente .pio/ ao seu .gitignore. É a pasta que guarda toolchain e objetos compilados, e ela cresce para centenas de megabytes.

4. Solução de problemas

Sintoma	Causa provável	O que fazer
LED_PIN não definido	compilou sem escolher ambiente	use -e esp32dev, -e esp32-c3 ou -e esp32-s3
Flag do [comum] sumiu	faltou referenciar o herdado	acrescente ${comum.build_flags} antes das suas flags
Primeira compilação demora muito	baixando toolchain	normal; as seguintes levam segundos
Mudei o .ini e nada mudou	artefatos antigos	rode pio run -t clean
Não acha a porta serial	driver USB-UART ausente	instale CH340, CP2102 ou FTDI conforme a placa
A placa não aparece	cabo só de carga	troque por cabo com transmissão de dados

5. Checklist final

	Item
[ ]	VS Code com a extensão PlatformIO IDE instalada
[ ]	Projeto aberto pela pasta, e não por arquivo solto
[ ]	Ambiente escolhido conforme a sua placa
[ ]	Pino de LED conferido no manual da sua placa
[ ]	Compilação concluiu sem erro
[ ]	Gravação concluiu e o LED pisca
[ ]	Monitor mostra o relato de estado a cada 10 s
[ ]	.pio/ acrescentado ao .gitignore
[ ]	Dependências com versão travada, se você acrescentou alguma