

Python no NuttX

Novas Possibilidades para Sistemas Embarcados

Tiago Medicci Serrano

2025-08-05

Apache NuttX

O que é o NuttX? 🤔

De <https://nuttx.apache.org/>:

NuttX is a real-time operating system (RTOS) with an emphasis on standards compliance and small footprint. Scalable from 8-bit to 64-bit microcontroller environments, the primary governing standards in NuttX are POSIX and ANSI standards. Additional standard APIs from Unix and other common RTOS's (such as VxWorks) are adopted for functionality not available under these standards, or for functionality that is not appropriate for deeply-embedded environments (such as `fork()`).



O que é o NuttX?

NuttX is a real-time operating system (RTOS).

Tal como outros sistemas operacionais (também RTOSes) conhecidos:

Sistema Operacional	Licença	Pulse/Authors ^[1] (Jul/2025)
Zephyr	Apache 2.0	359
ESP-IDF	Apache 2.0	76
NuttX	Apache 2.0	59
RIOT	LGPL2.1	19

1. *Pulse* é uma referência a repositórios públicos no GitHub. Entre outras informações úteis, o número de autores diferentes pode estar relacionado à atividade do projeto. ↩

O que é o NuttX?

(...) with an emphasis on standards compliance and small footprint. Scalable from 8-bit to 64-bit microcontroller environments (...)

Suporta múltiplas arquiteturas de hardware, de *Z80* à *RISC-Vs* de 64 bits e, claro, os seguintes SoCs da Espressif:

- Xtensa:
 - ESP32
 - ESP32-S2
 - ESP32-S3
- RISC-V:
 - ESP32-C3
 - ESP32-C6
 - ESP32-H2
 - ESP32-P4 (Em Breve)

O que é o NuttX?

(...) the primary governing standards in NuttX are POSIX and ANSI standards.

POSIX?

POSIX (um acrônimo para: Portable Operating System Interface, que pode ser traduzido como Interface Portável entre Sistemas Operativos) é uma família de normas definidas pelo IEEE para a manutenção de compatibilidade entre sistemas operacionais (sistemas operativos em PT-PT), e designada formalmente por IEEE 1003. POSIX define a interface de programação de aplicações (API), juntamente com shells de linha e comando e interfaces utilitárias, para compatibilidade de software com variantes de Unix e outros sistemas operacionais.

De <https://pt.wikipedia.org/wiki/POSIX>, acessado em 23 de julho de 2025.

Sim, POSIX! 🙌

Ainda de <https://pt.wikipedia.org/wiki/POSIX>, acessado em 23 de julho de 2025:

Tem como objetivo garantir a portabilidade do código-fonte de um programa a partir de um sistema operacional que atenda às normas POSIX para outro sistema POSIX, desta forma as regras atuam como uma interface entre sistemas operacionais distintos.

POSIX: *O Super Trunfo do NuttX!* 🖊️

Reinventing the wheel ❌

Alguém já teve que fazer quase a mesma aplicação que você!

Code reuse ✅

Provavelmente, esta aplicação já foi desenvolvida para sistemas operacionais Unix com interfaces POSIX 😊

O NuttX é o RTOS mais próximo do Linux embarcado para microcontroladores!

e essa afirmação *esconde* uma série de vantagens...

Hello, World!

Quais são as interfaces POSIX?

```
#include <nuttx/config.h>
#include <stdio.h>

int main(int argc, FAR char *argv[])
{
    printf("Hello, World!!\n");
    return 0;
}
```

The Open Group Base Specifications Issue 8
IEEE Std 1003.1-2024
Copyright © 2001-2024 The IEEE and The Open Group

NAME

stdio.h — standard buffered input/output

SYNOPSIS

```
#include <stdio.h>
```

The Open Group Base Specifications Issue 8
IEEE Std 1003.1-2024
Copyright © 2001-2024 The IEEE and The Open Group

NAME

printf — write formatted output

SYNOPSIS

```
printf format [argument...]
```

Hello, World!, ELF loader, Web-assembly e muito mais...

A apresentação do NuttX no Espressif Summit Brasil 2024 está disponível em:

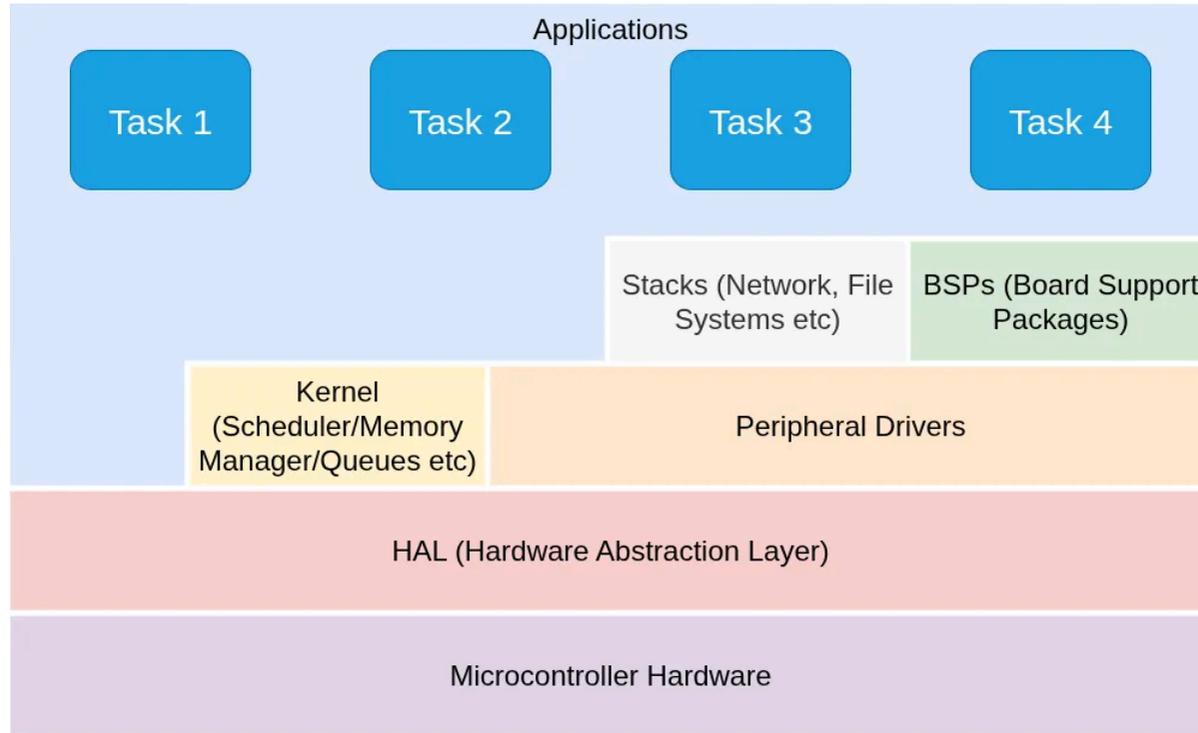
<https://tmedicci.github.io/esb24-nuttX>

Ok, o NuttX é compatível com POSIX, o acesso ao hardware é feito através de interfaces conhecidas e aplicações podem ser compiladas externamente e carregadas no firmware, mas...

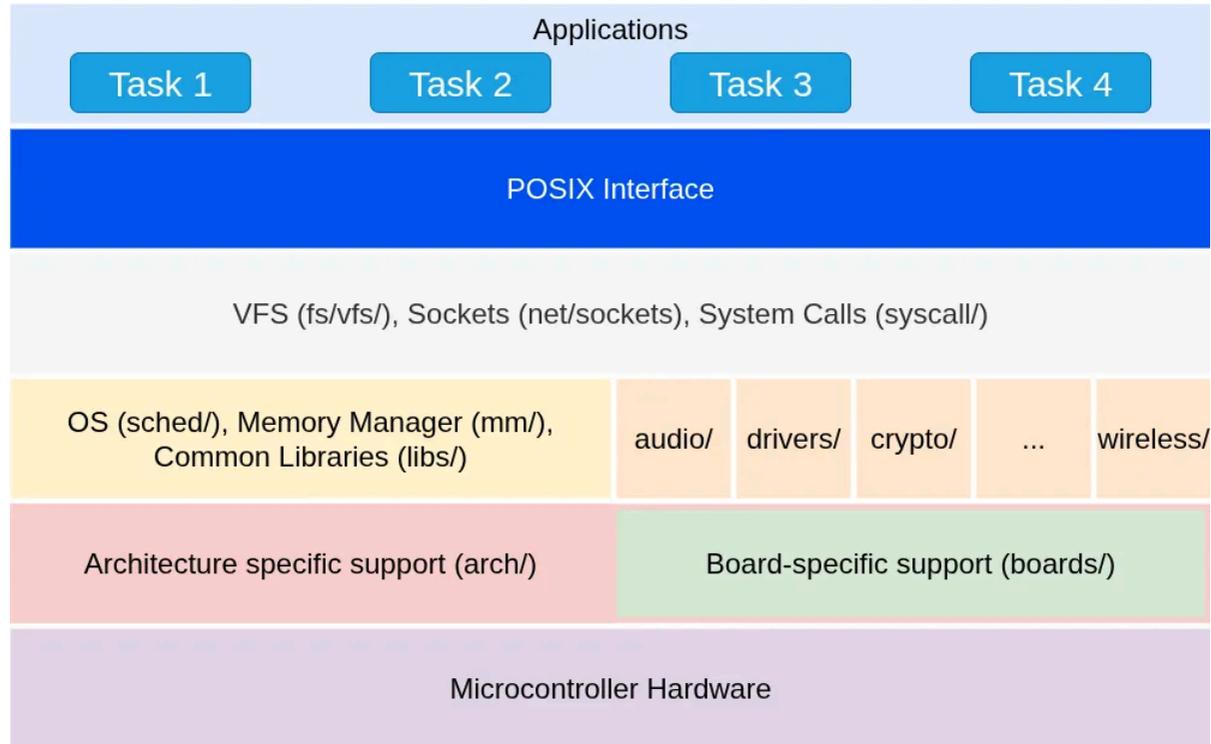
Por que usar o NuttX?

Plataforma Computacional Embarcada 

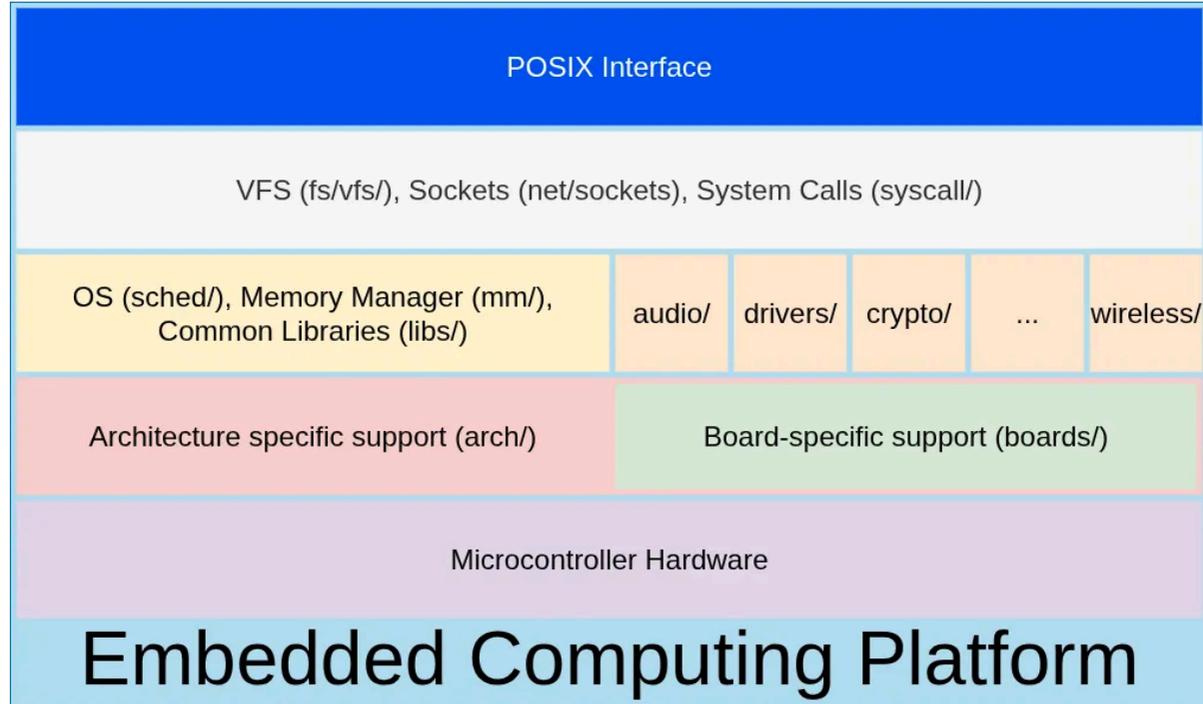
Arquitetura Tradicional de Sistemas Operacionais Embarcados



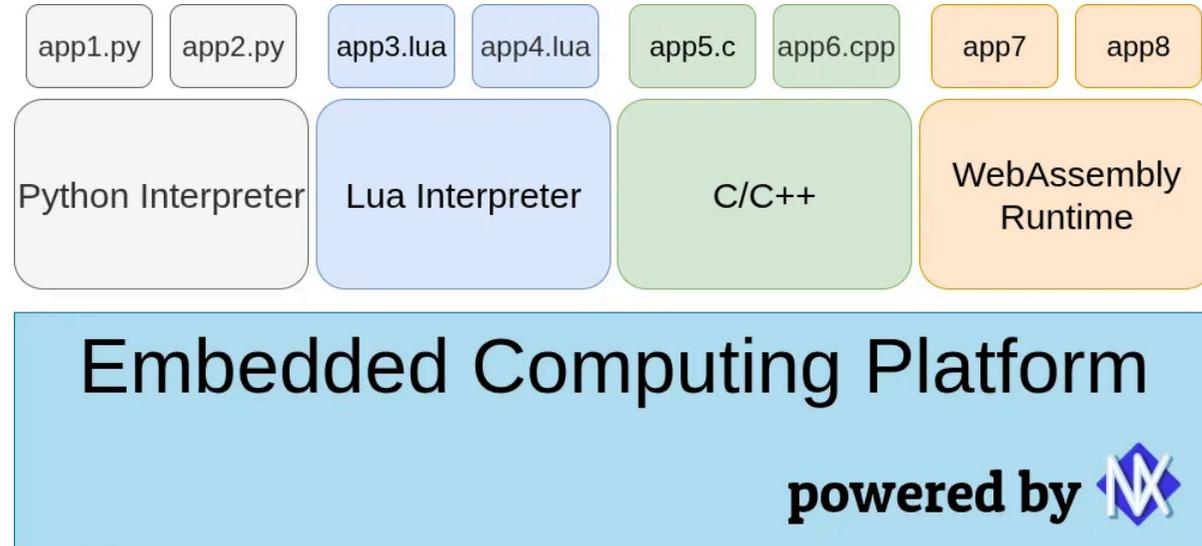
e no NuttX...



E, deixando as aplicações de lado por enquanto...



Ou, simplificando...



Novos Conceitos

Sistema Embarcado vs Plataforma Computacional :

- Interface padrão para acesso ao hardware (POSIX):
 - Reutilização de Código
 - Múltiplas linguagens de programação
 - Controle de acesso aos periféricos controlado pelo sistema operacional
- Base para o desenvolvimento de diferentes produtos
- Atualização de Sistema vs Atualização de Aplicações
- Democratização da Programação Embarcada:
 - Engenheiro de Sistemas Embarcados: plataforma computacional
 - Desenvolvedores de Software: desenvolvimento de aplicações

E, finalmente...

Python no NuttX

Por que Python?

- De IEEE Spectrum's 11th annual rankings^[1]:
 - Linguagem mais adotada entre os membros do IEEE
 - O dobro do segundo colocado (Java)
- De 2024 Stack Overflow Developer Survey^[2]:
 - Usada por mais de 51% dos desenvolvedores

 *O Python é a linguagem de programação mais usada no mundo, considerando as linguagens não específicas para Web.*

1. <https://spectrum.ieee.org/top-programming-languages-2024> ←

2. <https://survey.stackoverflow.co/2024/technology/#1-programming-scripting-and-markup-languages> ←

Python no NuttX

Por que Python (e não MicroPython)?

Footprint

Aspecto	Python 	MicroPython 
Uso de Memória	MiB	KiB
Tamanho	Dezenas de MiB	< MiB
Dependências	POSIX	Bare-metal/RTOS-dependent

Por que Python (e não MicroPython)?

Gerenciamento de Memória

Aspecto	Python 	MicroPython
Garbage Collection	Automático	Manual
Alocação de Memória	Dinâmico/Flexível	Limitada (requer refinamento)
Esforço de Desenvolvimento	Menor	Maior (requer gerenciamento)

Python no NuttX

Por que Python (e não MicroPython)?

Bibliotecas

Aspecto	Python 	MicroPython
Abrangência	Python's Standard Library	Subconjunto (necessário para dispositivos embarcados)
Acesso ao Hardware	Interfaces POSIX (arquivos e sockets)	Direto

Python no NuttX 

Por que Python (e não MicroPython)?

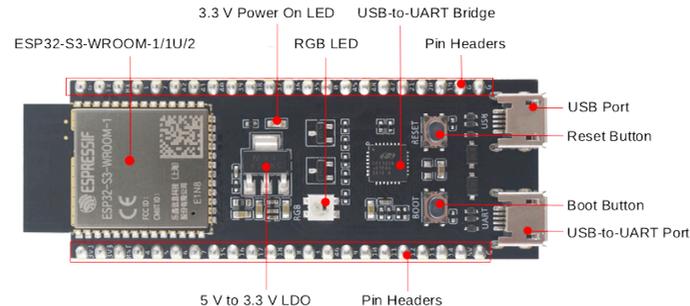
Porque podemos!

Python no NuttX

ESP32-S3

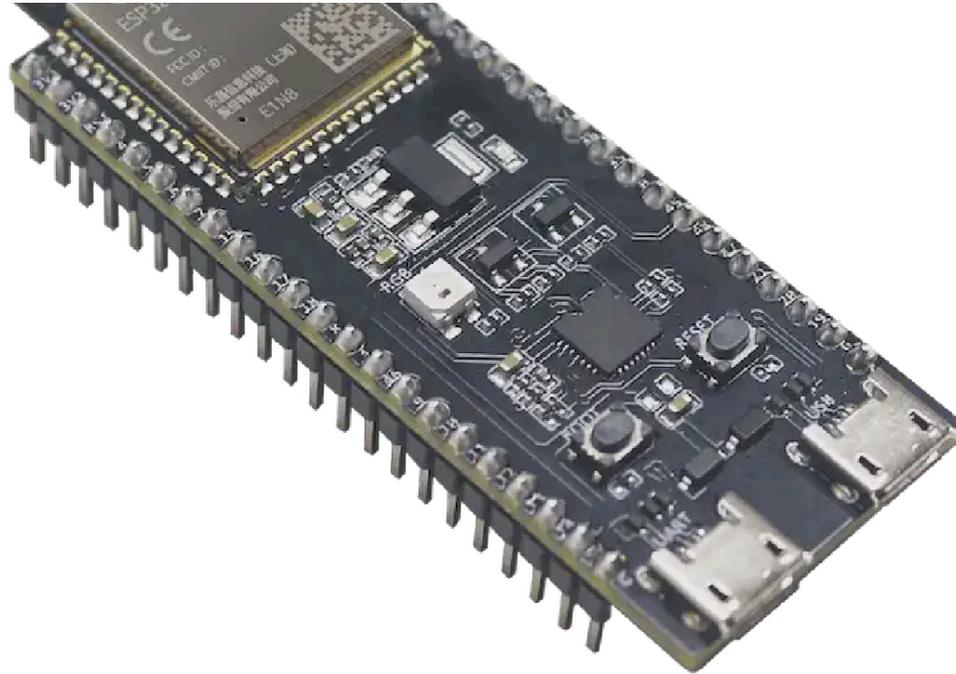
- Ao menos 8MiB de memória RAM externa (PSRAM)
- Ao menos 16MiB de memória Flash

O Módulo `ESP32-S3-WROOM-2-N32R8V` na placa `ESP32-S3-DevKitC-1` ^[1] cumpre estes requisitos da placa, por exemplo.



1. <https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32s3/esp32-s3-devkitc-1/index.html> ↩

De [https://developer.espressif.com/blog/2025/03/nuttx-python-esp32s3:](https://developer.espressif.com/blog/2025/03/nuttx-python-esp32s3)



Exemplo: LED endereçável da Placa `ESP32-S3-DevKitC-1` 



Exemplo: LED endereçável da Placa ESP32-S3-DevKitC-1



Aplicação `ws2812` , em https://github.com/apache/nuttx-apps/blob/master/examples/ws2812/ws2812_main.c:

```
1  free(buffer);
2  close(fd);
3  fflush(stdout);
4  return OK;
5
6  errout_with_dev:
7  close(fd);
8
9  errout:
10 fflush(stdout);
11 return ERROR;
12 }
```

Exemplo: LED endereçável da Placa `ESP32-S3-DevKitC-1` 



Desafio: Como fazer uma aplicação que monitora o uso da CPU e, de acordo com o percentual de tempo ocioso da CPU, acende o LED endereçável em uma escala de verde (totalmente ocioso) para vermelho (totalmente ocupada) em Python? 

Perguntar para o ChatGPT, claro!

ou qualquer outra LLM, *pick yours!*

Exemplo: LED endereçável da Placa ESP32-S3-DevKitC-1



Indicando o dispositivo referente ao LED endereçável (`/dev/leds0`), o arquivo que mostra o uso da CPU (`/proc/cpuload`) e o código do programa `ws2812_main.c` como exemplo, a LLM foi capaz de gerar o seguinte arquivo:

`cpumon.py` :

```
# Write to device
try:
    with open('/dev/leds0', 'wb') as f:
        f.write(data)
except IOError as e:
    print(f"Error writing to device: {e}")
    sys.exit(1)

# Wait 100ms using select (instead of time.sleep)
select.select([], [], [], 0.1)

except KeyboardInterrupt:
    print("\nExiting... ")
    sys.exit(0)

if __name__ == '__main__':
    main()
```

Exemplo: LED endereçável da Placa `ESP32-S3-DevKitC-1` 



Ok, mas como enviar e executar este script no NuttX?

- Gravando em um cartão SD
- Gravando em uma partição da Flash
 - Converter o arquivo para um sistema de arquivos suportado (`SPIFFS` , `smartFS` , `FAT`)
 - Mover para o cartão SD/gravar na memória Flash em uma posição conhecida
 - Montar o cartão SD/memória Flash no NuttX
- Transferir pelo Wi-Fi para uma partição/sistema de arquivos já existente

Interfaces Não Padronizadas 🗝️

Ah, mas e se não existir uma interface POSIX, como o script Python interage com o NuttX?

O Gerenciador de Conexões de Rede

- `NetworkManager`
- `ConnMan`
- `WICD`
- `WAPI` (NuttX)

A Solução: `import os`

```
import os

# Qualquer comando ou aplicação disponível no NuttShell (`nsh > help`) pode ser executado com `os.system`

os.system("wapi --help")
```

Exemplo: Conectando-se ao Wi-Fi via `wapi`



Referências

- Esta apresentação: <https://tmedicci.github.io/esb25-nuttX/>
- Running Python on ESP32-S3 with NuttX^[1]
- Apache NuttX: Porting Python to NuttX^[2]
- NuttX Documentation: `python` interpreter^[3]

Onde me encontrar?

  @tmedicci

Questões?

1. <https://developer.espressif.com/blog/2025/03/nuttX-python-esp32s3/> ←
2. https://tmedicci.github.io/articles/2025/01/08/python_on_nuttX.html ←
3. <https://nuttx.apache.org/docs/latest/applications/interpreters/python/index.html> ←