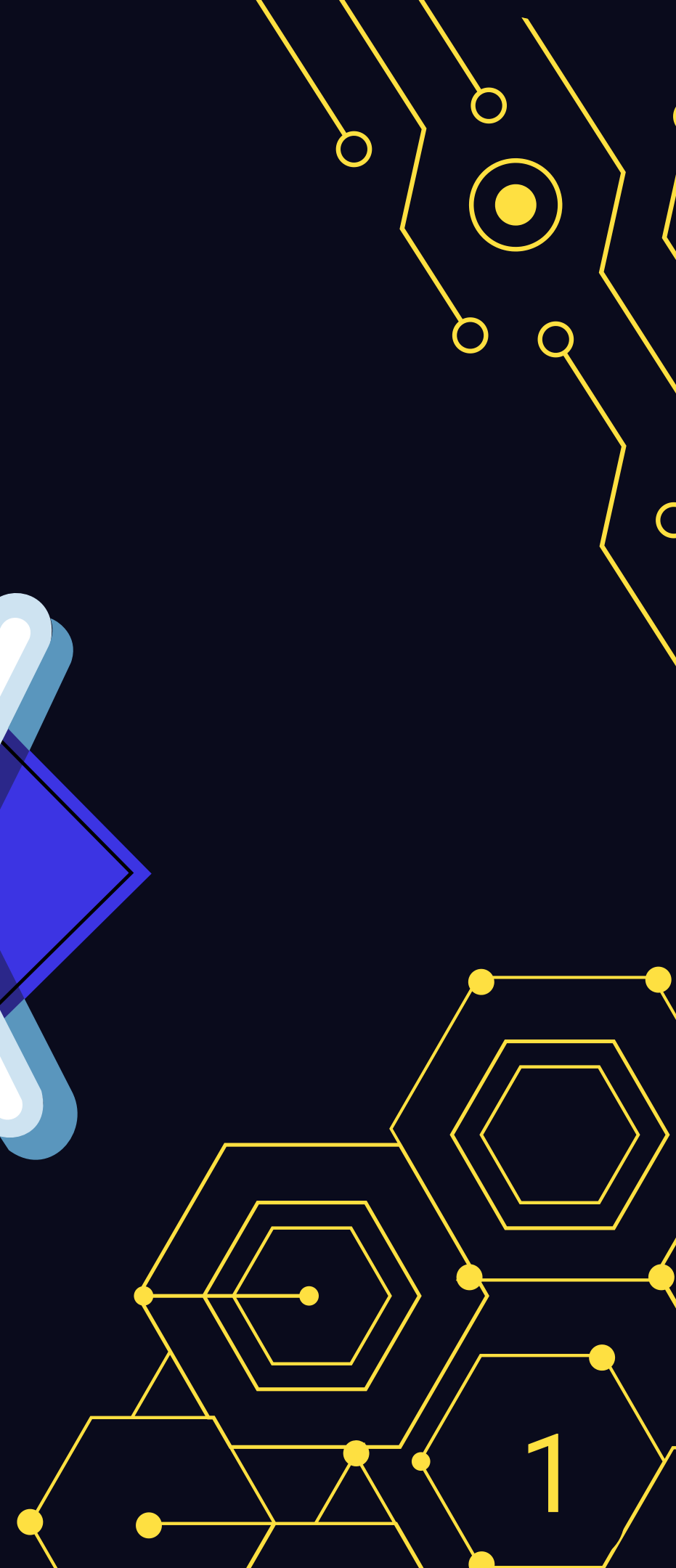


Sistema Operacional Embarcado

NuttX RTOS

Paulo Diego De Meneses
Felipe de Sousa Alves

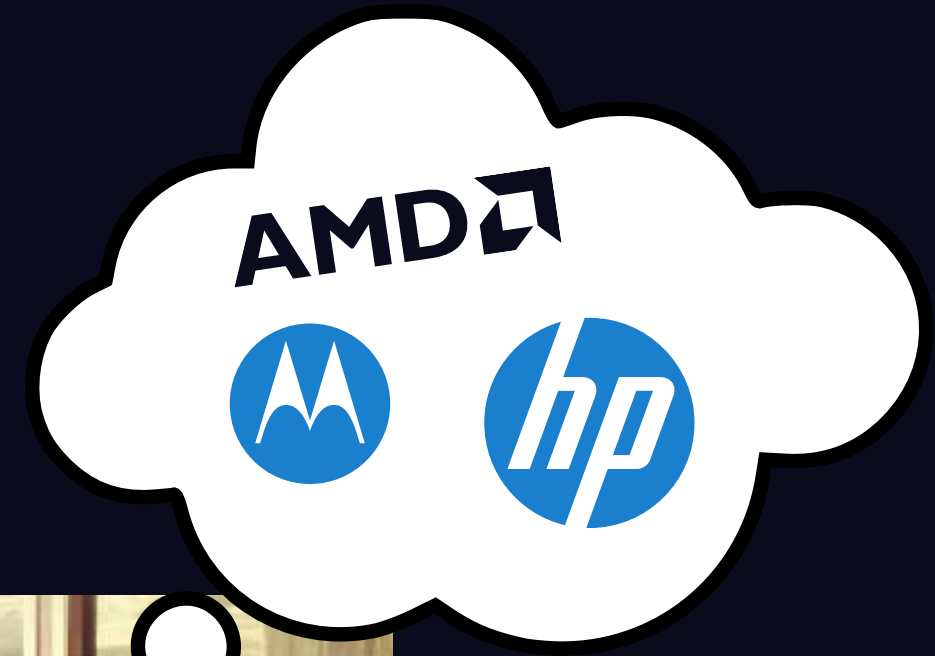


O que é o Nuttx?

Nuttx

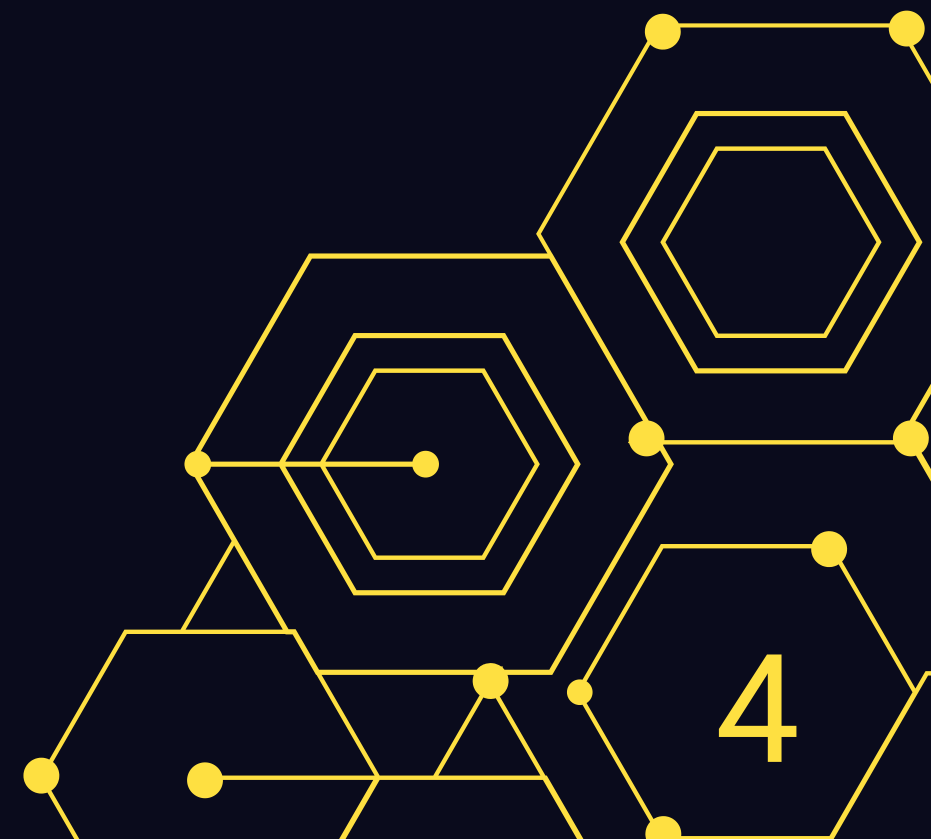
É um RTOS criado por Gregory Nutt.

- Lançado em Fevereiro de 2007
- Pequenas pegadas
- Conjunto de funcionalidades
- Escalável
- Padronização
 - POSIX
 - ANSI
- Licenciamento Apache 2.0
- Escrito em C, C++, Assembly
- Último release: 30/10/2024
 - v12.7.0



Algumas funcionalidades

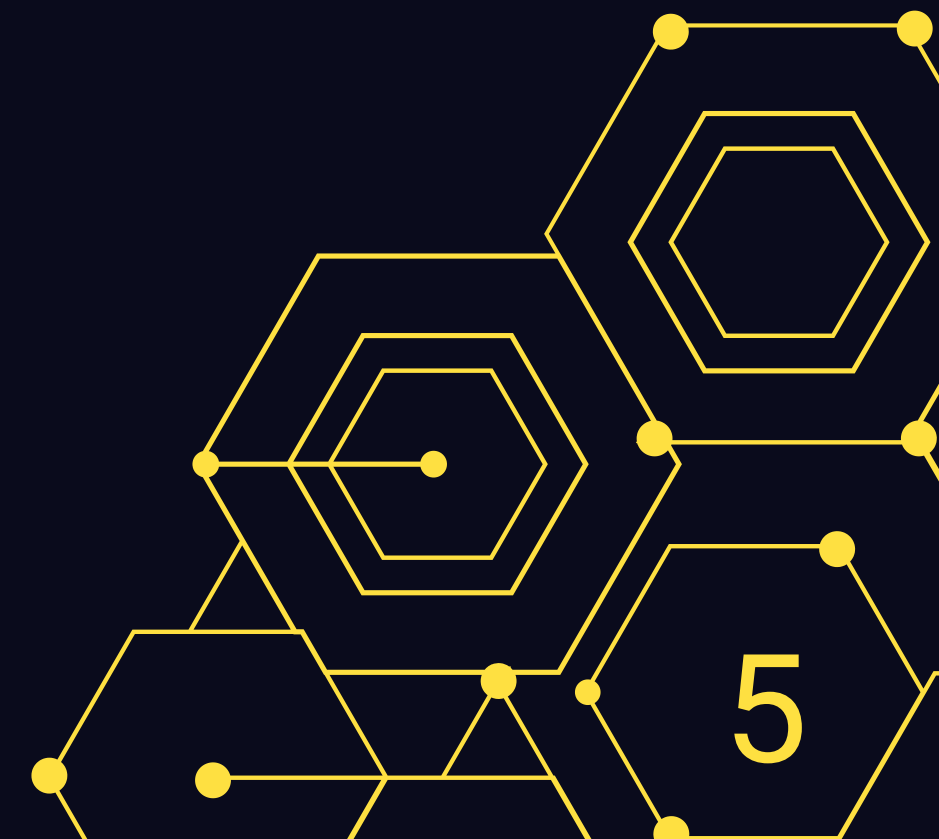
- Gerenciamento de Memória
- Sistema de Arquivos
- Redes
- Baixa Pegada de Memória (Small Footprint)
 - Bibliotecas Estáticas: Muitos arquivos-fonte pequenos, geralmente com apenas uma função.
 - Arquivos de Configuração: Um arquivo de configuração define quais recursos serão usados.
- Drivers e Suporte de Hardware.
- Compatibilidade com ferramentas GNU



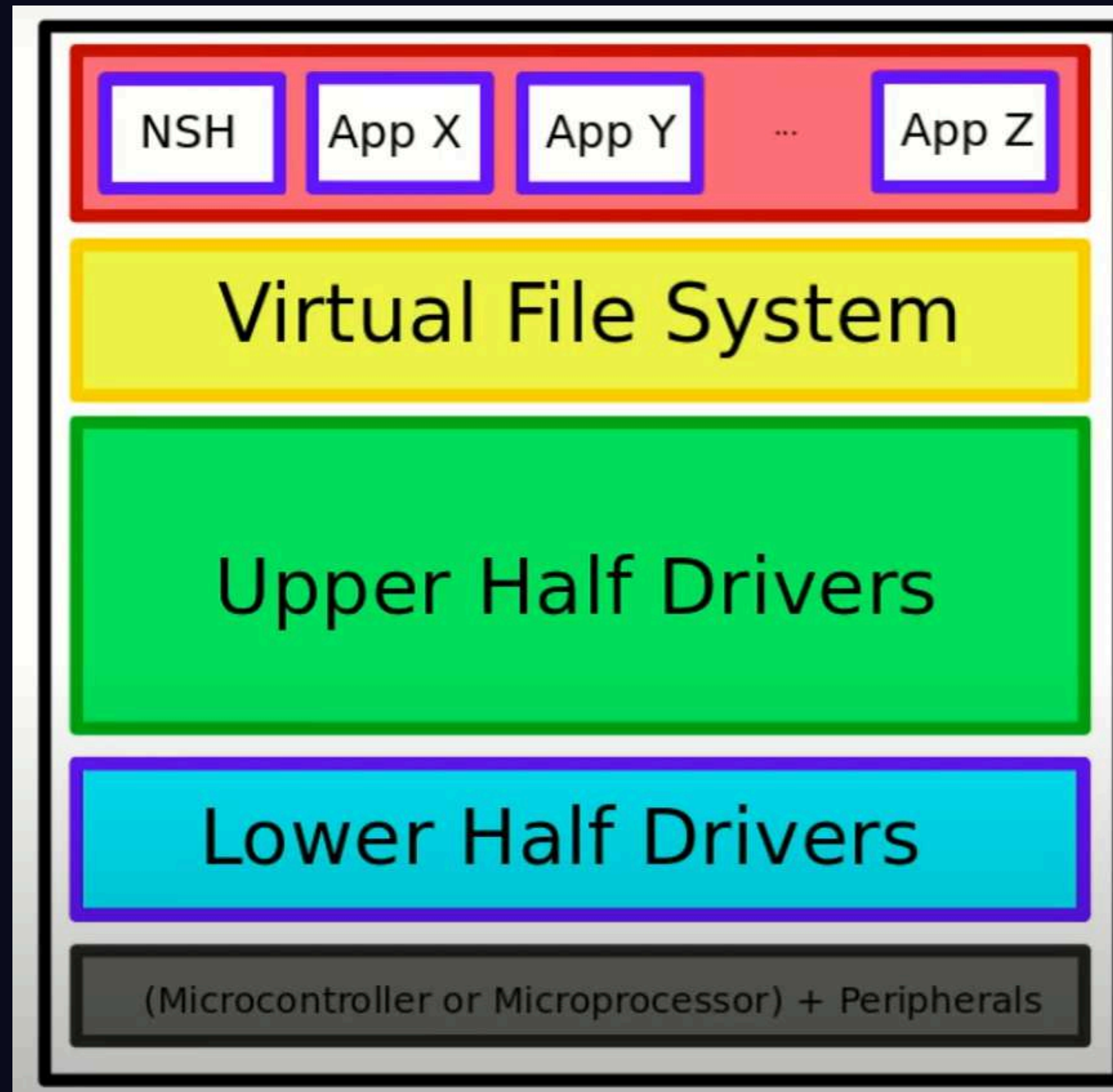
Estrutura do Nuttx

Dividido em subsistemas modulares, cada um com suas bibliotecas.

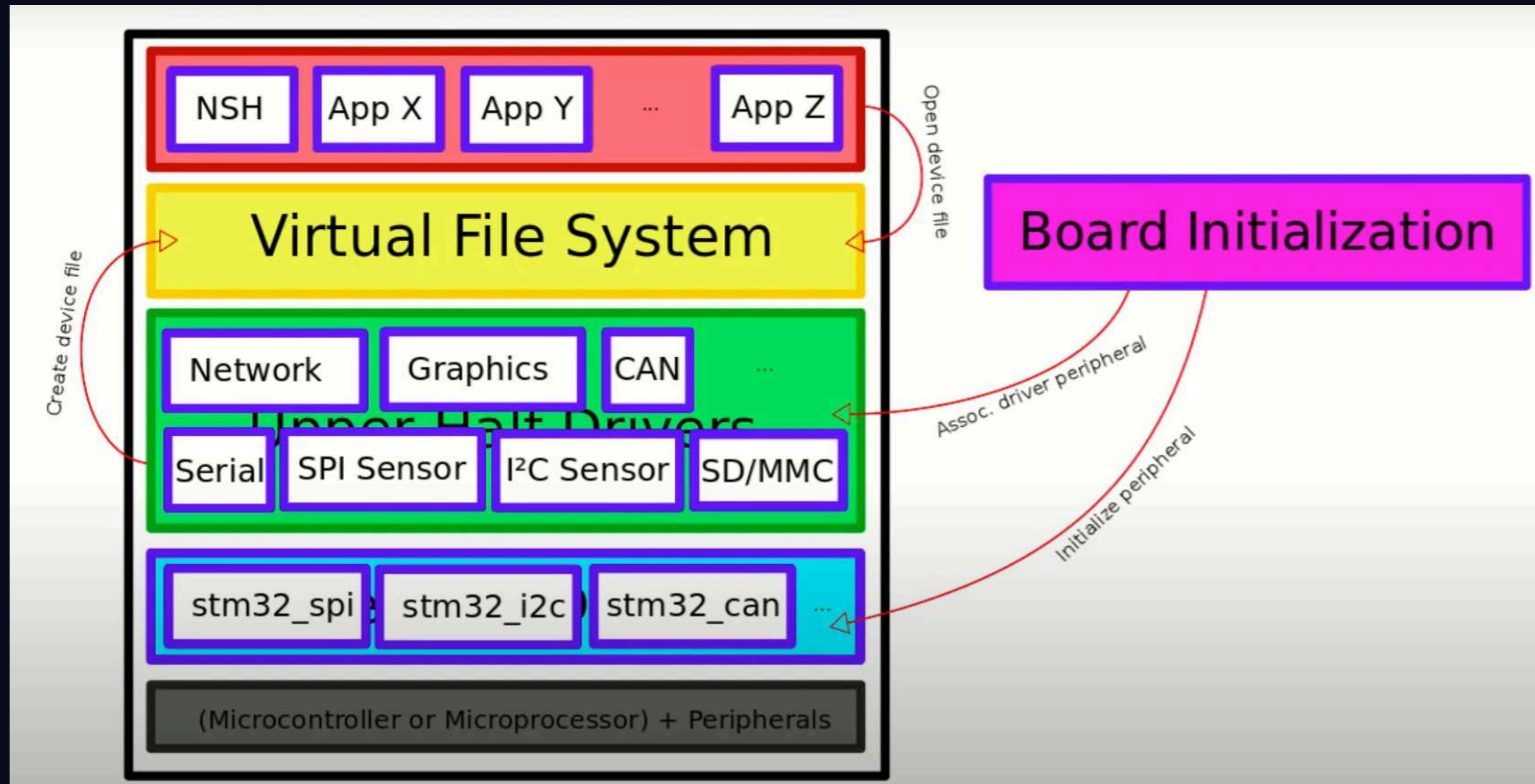
- Carregador de Binário
- Executador de programas
- Drivers
- Redes
- Gráfico (imagens, vídeos)
- Áudio
- Sistema de arquivos (ex.: FAT e NFS)
- Gerenciamento de memória
- Chamadas de sistema (POSIX)
- Portabilidade (ARM, ESP32, RISC-V, etc.)
 - Código de arquitetura específica



Organização dos drivers

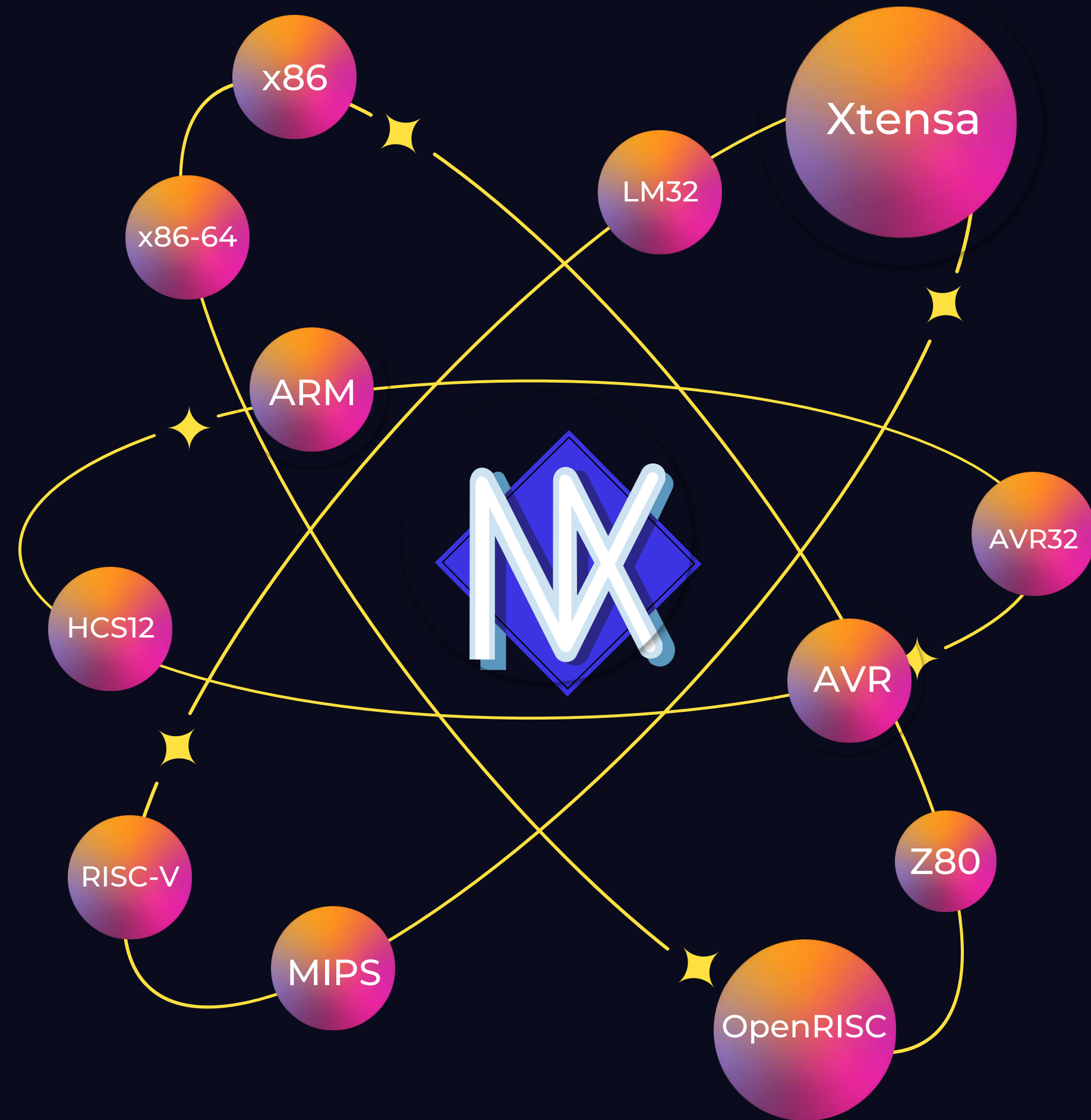


Organização dos drivers



Plataformas Compatíveis



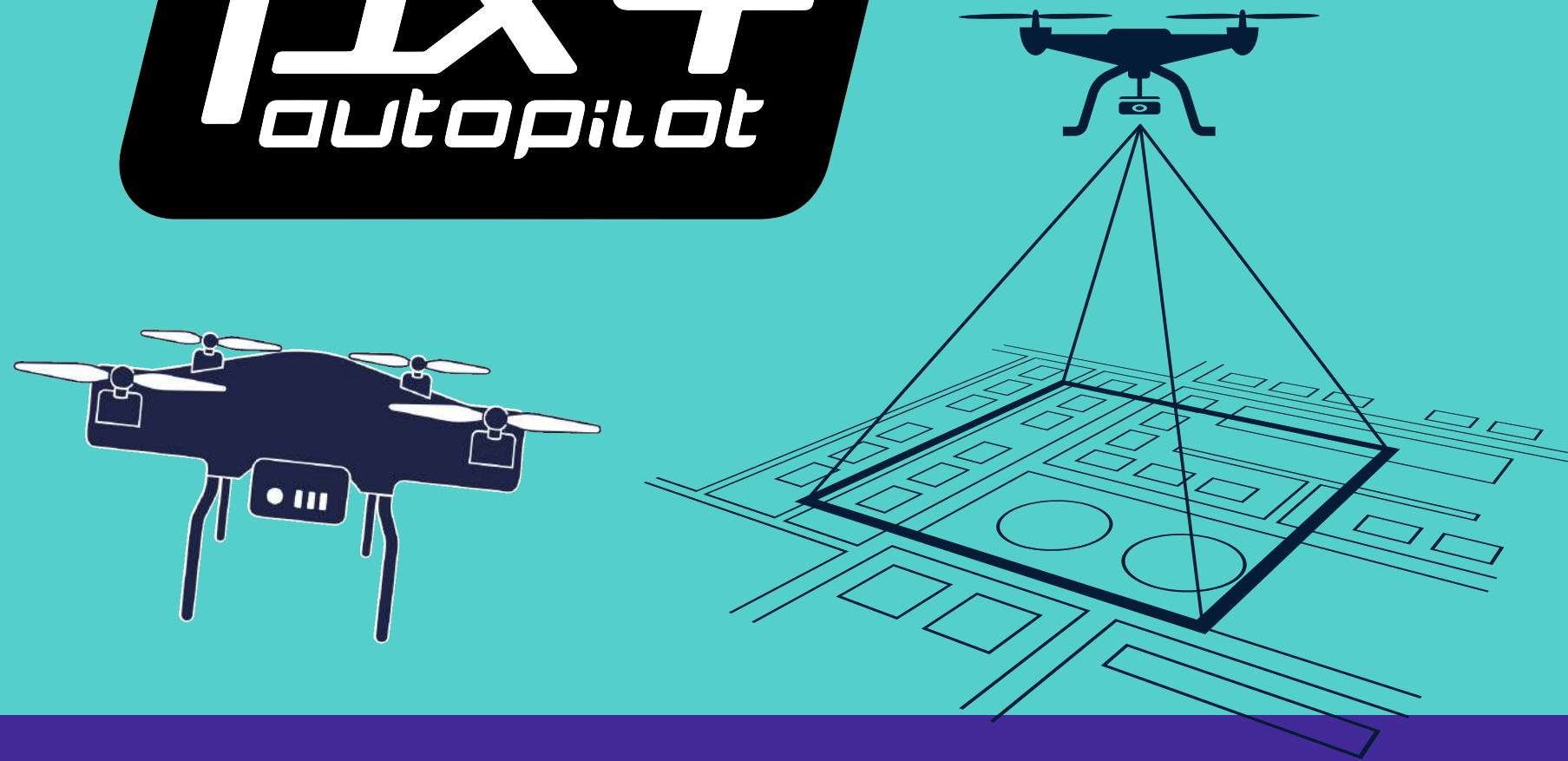


Plataformas

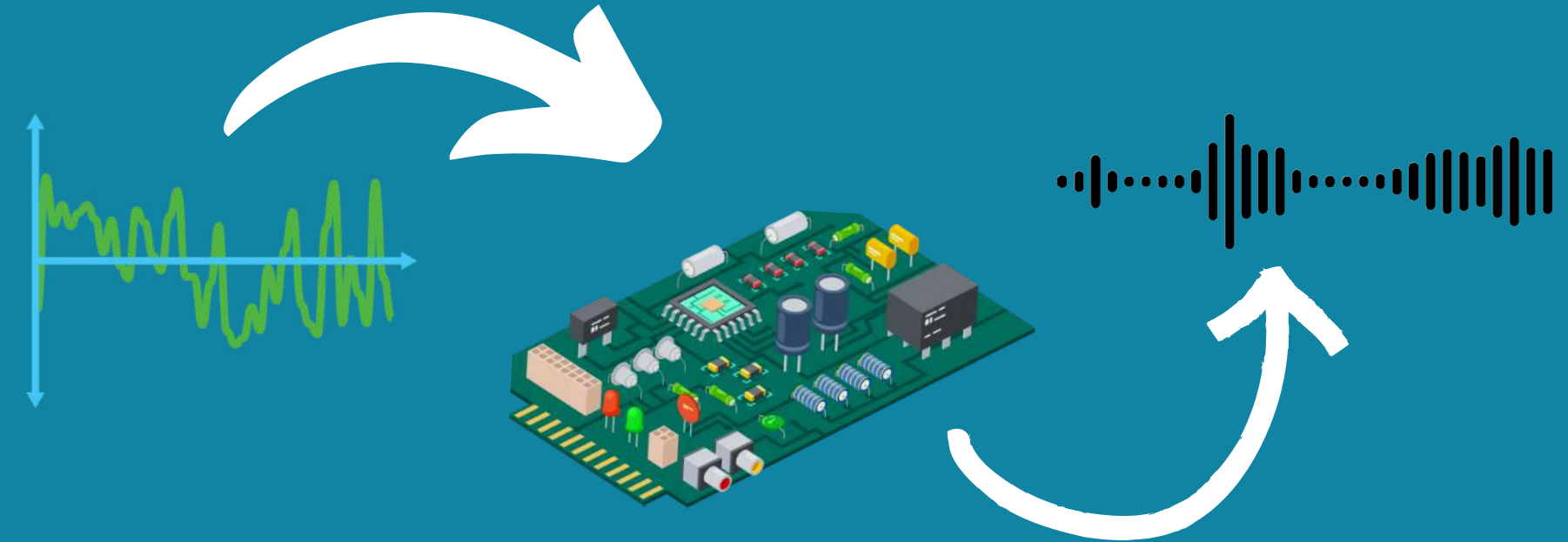


Nuttix no mercado

PX4
autopilot



SONY



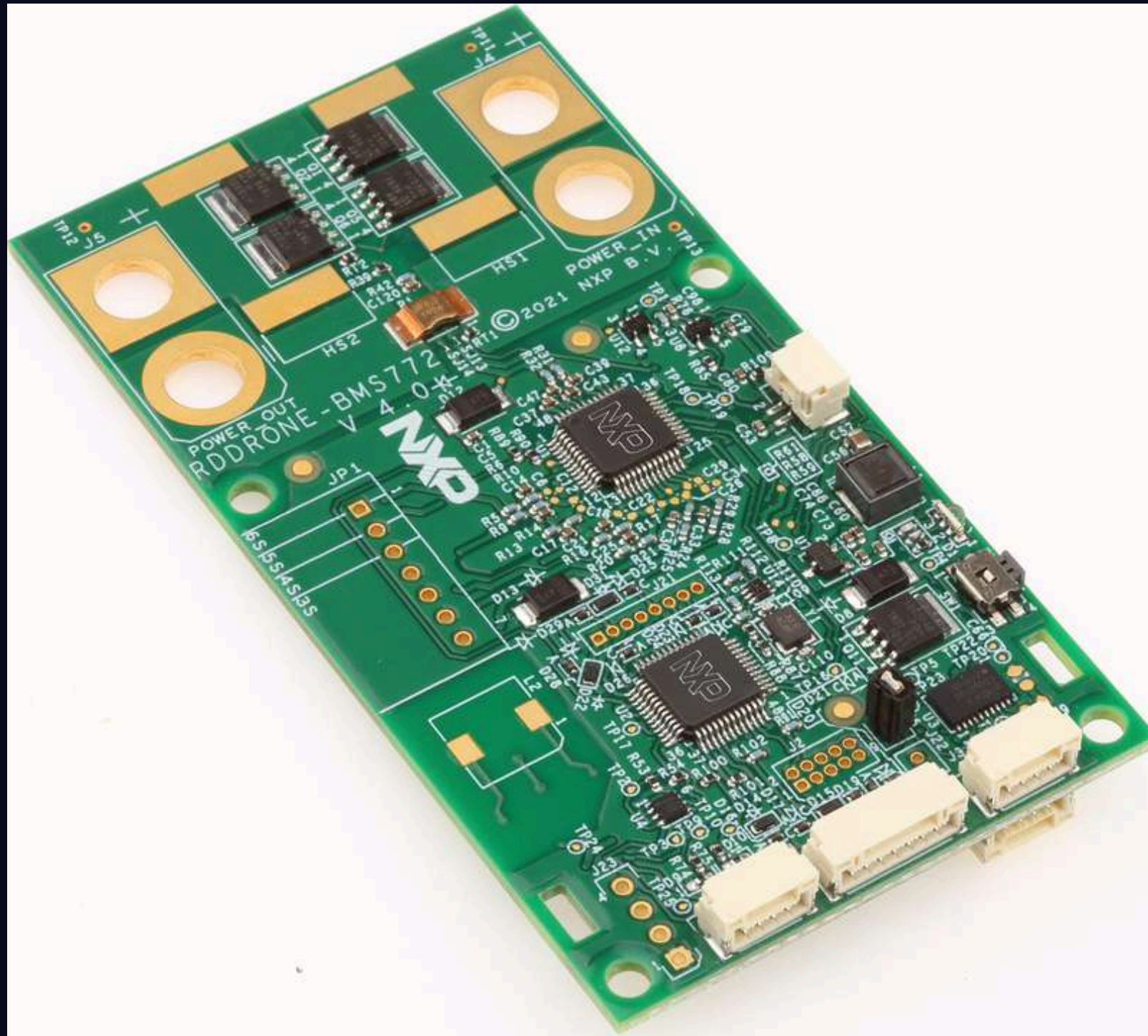
SAMSUNG



MOTOROLA

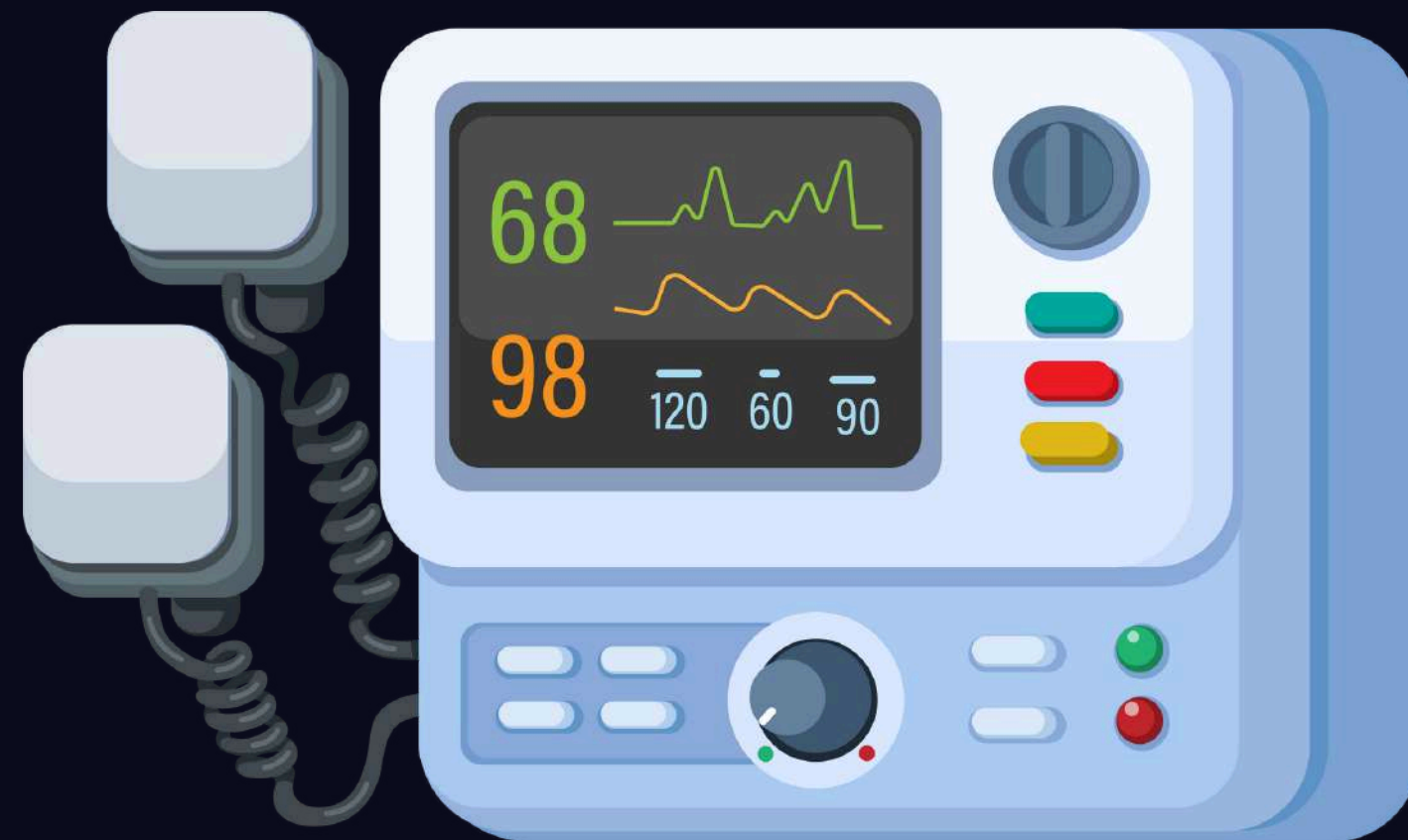


RDDRONE-BMS772



- Sistema de Gerenciamento de Bateria
- NXP MC33772
 - 7 ADC/GPIO/Temperature sensor inputs
- S32K146/144
 - Arm Cortex-M4F
 - 32 bits

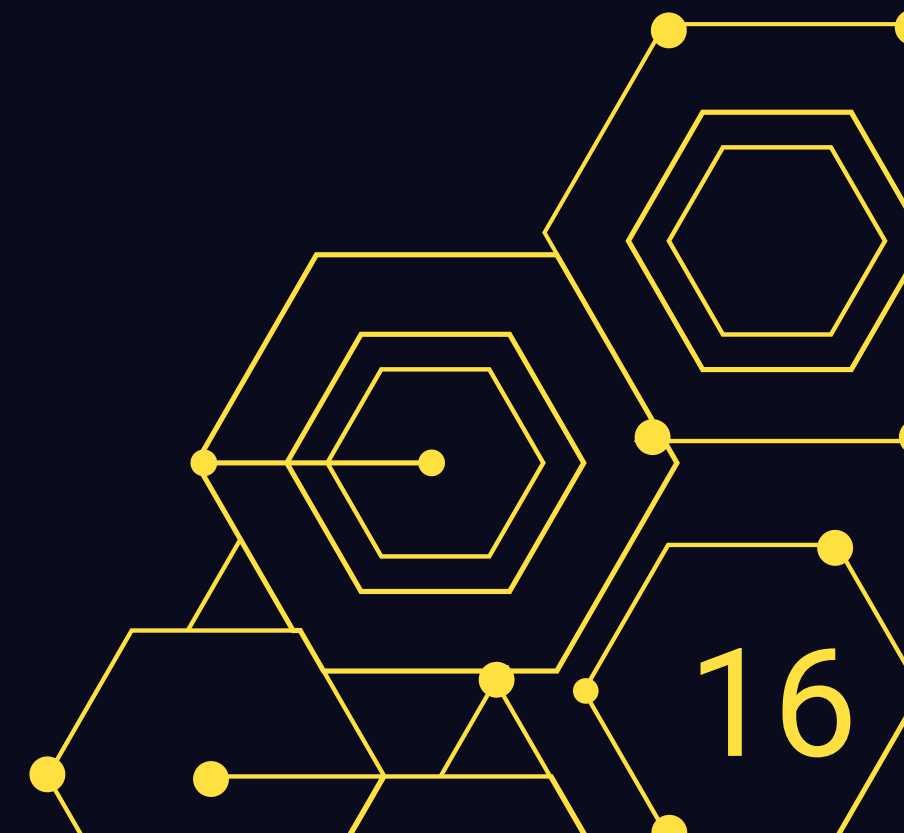
Microcontrolador só para a bateria?



Como Usar

Ambiente de desenvolvimento

- Pode ser Linux, Windows, MacOS
- Precisa ter GNU make, bash, e ferramentas linux (cat, etc)
 - O GNU Make é uma ferramenta que controla a geração de executáveis.



Sequência de uso

- Instalar Pacotes básicos do sistema
- Compilador Cruzado
- Kernel do Nuttx e Apps
- Bootloader e o partition
- Ferramentas adicionais
 - Ex: esptool
- Compila e grava na placa
- Monitora
 - Via NSH

```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3ffb2140,len:1760
load:0x40080000,len:20664
entry 0x4008260c
*** Booting NuttX ***
dram: lma 0x00001020 vma 0x3ffb2140 len 0x6e0 (1760)
iram: lma 0x00001708 vma 0x40080000 len 0x50b8 (20664)
padd: lma 0x000067d8 vma 0x00000000 len 0x9820 (38944)
imap: lma 0x00010000 vma 0x400e0000 len 0x132e4 (78564)
padd: lma 0x000232ec vma 0x00000000 len 0xcd2c (52524)
dmap: lma 0x00030020 vma 0x3f400020 len 0x3314 (13076)
total segments stored 6
WARNING: NuttX supports ESP32 chip revision >= v3.0 (chip is v1.0).
Ignoring this error and continuing because `ESP32_IGNORE_CHIP_REVISION_CHECK` is set...
THIS MAY NOT WORK! DON'T USE THIS CHIP IN PRODUCTION!

NuttShell (NSH) NuttX-12.7.0
nsh> ?
```

Compilando...

```
# !/bin/bash
# A simple bash script to build the NuttX RTOS
#export ESP32 env
. $HOME/esp/esp-idf/export.sh
if [ $1 -eq 1 ]
then
    #distclean
    echo "Dist clean"
    make distclean

    #ESP32 configure
    echo "configuring ESP32 with NuttX RTOS"
    ./tools/configure.sh esp32-devkitc:nsh
fi
#menuconfig
make menuconfig
#make and flash
make download ESPTOOL_PORT=/dev/ttyUSB0 ESPTOOL_BINDIR=./esp-bins EXTRAFLAGS="-DESP32_IGNORE_CHIP_REVISION_CHECK"
```

Menu de configurações

.config - NuttX/x86_64 Configuration

NuttX/x86_64 Configuration

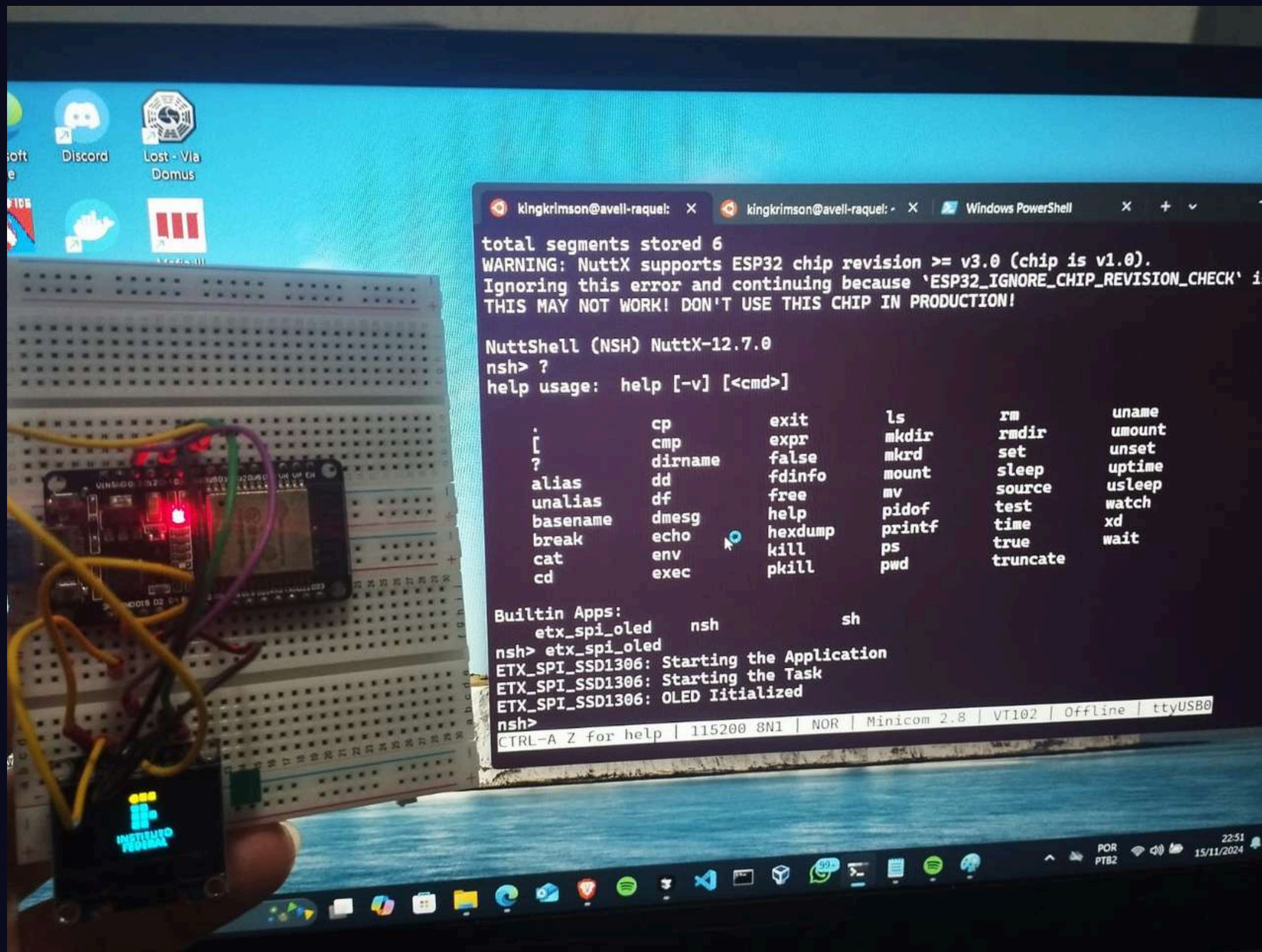
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ---). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module < > module capable

```
License Setup --->
Build Setup --->
System Type --->
Board Selection --->
RTOS Features --->
Device Drivers --->
Networking Support --->
Crypto API --->
File Systems --->
Graphics Support --->
Memory Management --->
```

↓(+)

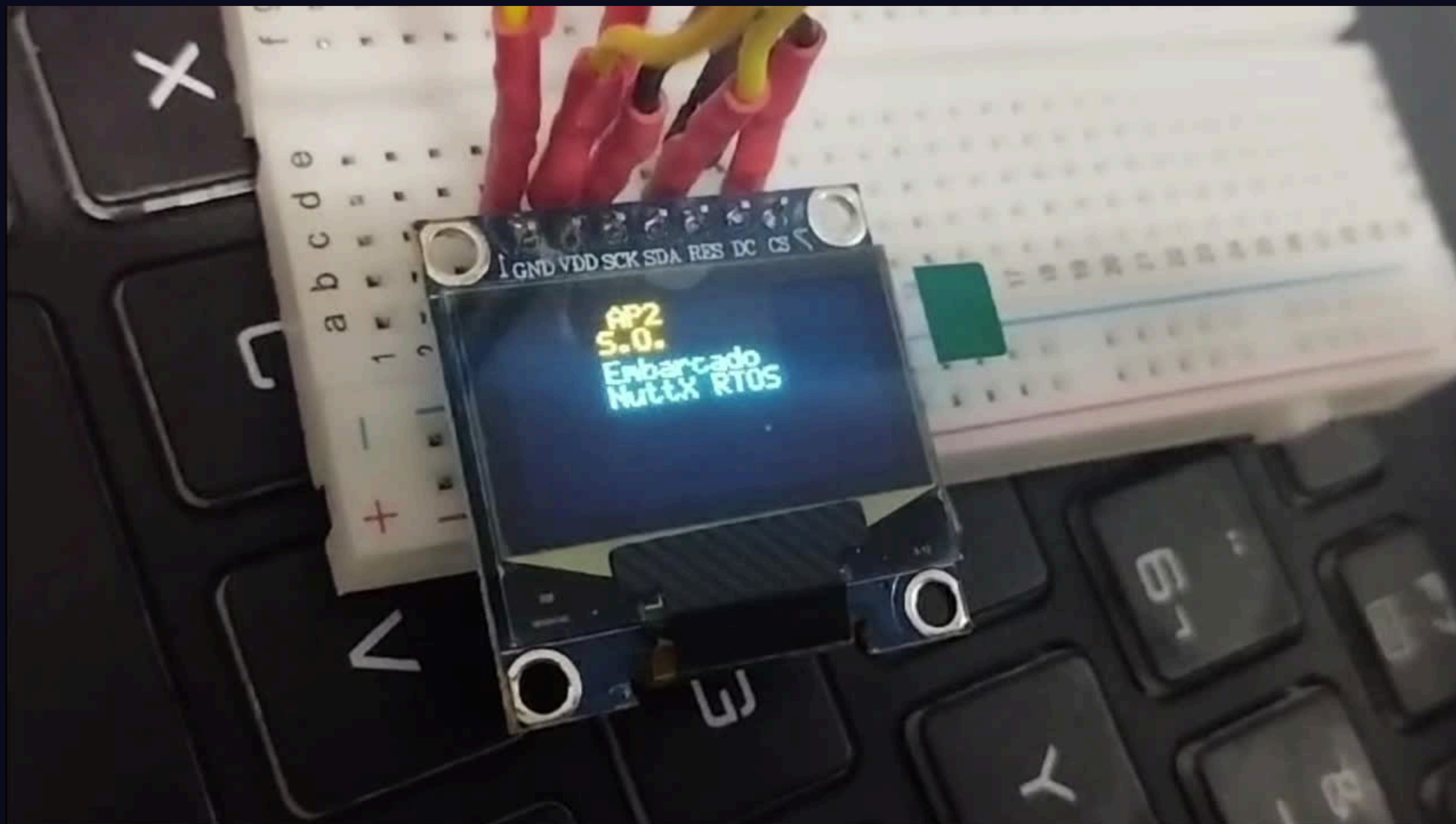
<Select> < Exit > < Help > < Save > < Load >

Exemplo usando ESP32



Tudo começa aqui!

```
ret = task_create( "ETX_SPI_SSD1306",           // Task Name
                  CONFIG_EXAMPLES_ETX_SPI_SSD1306_APP_PRIORITY, // Task priority
                  CONFIG_EXAMPLES_ETX_SPI_SSD1306_APP_STACKSIZE, // Task Stack size
                  etx_spi_ssd1306_task,          // Task function
                  NULL                           // Task parameters
                );
```



Bibliografia

- Site Oficial
 - <https://nuttx.apache.org/docs/latest/index.html>
- NuttX + Arduino Framework on ESP32
 - <https://www.industrialshields.com/blog/arduino-industrial-1/first-steps-with-apache-nuttx-rtos-and-esp32-plc-571>
- Getting Started with ESP32 and NuttX
 - <https://medium.com/the-esp-journal/getting-started-with-esp32-and-nuttx-fd3e1a3d182c>
- Site da Embarcados
 - <https://embarcados.com.br/nuttx-criando-ou-copiando-uma-aplicacao-para-o-nuttx/>
 - <https://embarcados.com.br/o-que-e-o-rtos-nuttx/>
- RDDRONE-BMS772 for Mobile Robotics
 - <https://nxp.gitbook.io/rddrone-bms772/software-guide-nuttx/untitled>
- Wikipedia
 - <https://en.wikipedia.org/wiki/NuttX>
- Canal da Nuttx Brasil
 - <https://www.youtube.com/@NuttXBrasil>
- Canal da EmbeTronicX
 - <https://www.youtube.com/@EmbeTronicX>

Perguntas?

Obrigado!