Introduction to Apache NuttX

Who am I?

Eren Terzioğlu

Computer Engineer from Yildiz Technical University. Software engineer at Espressif Systems. Maker, Electronics Hobbyist

/erenterzioglu
/eren-terzioglu

/eren-terzioglu

What is NuttX?

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.

- Source: Apache NuttX

What is RTOS?

A **real-time operating system** (**RTOS**) is an operating system (OS) for real-time computing applications that processes data and events that have critically defined time constraints. All operations must verifiably complete within given time and resource constraints or else fail safe. Real-time operating systems are event-driven and preemptive, meaning the OS can monitor the relevant priority of competing tasks, and make changes to the task priority. Event-driven systems switch between tasks based on their priorities, while time-sharing systems switch the task based on clock interrupts.

- Source Real-time operating system - Wikipedia

Operating system	First commit	Governance	License	Contributors	Pulse (jun10/2024)
Zephyr	2014	community	Apache 2.0	100+	942
NuttX	2007	community	Apache 2.0	100+	135
RT-Thread	2009	community	Apache 2.0	100+	67
RIOT	2010	community	LGPL2.1	100+	71
Tyzen RT	2015	Samsung	Apache 2.0	100+	36
myNewt	2015	Community	Apache 2.0	100+	25
mbed OS	2013	ARM	Apache 2.0 or BSD-3 Clause	100+	7
FreeRTOS	2004	Richard Barry	MIT	100+	6
Contiki-NG	2016	community	BSD-3 Clause	100+	4
CMSIS-5	2016	ARM	Apache 2.0	100+	0
Azure-RTOS	2020	Microsoft	Microsoft Software License	10+	archived

NuttX is the 2nd most popular community-based RTOS (along with Zephyr in the 1st position):

— Table by Alin Jerpelea, presented on NuttX Workshop 2024

Advantages of NuttX

- Apache 2.0 Licensed
- Small footprint
- Variety of architecture support (from Z80 to 64 bit RISC-V)
- Community support
- POSIX compliant

arm	risc-v
arm64	sim
avr	sparc
ceva	tricore
dummy	x86
hc	x86_64
mips	xtensa
misoc	z16
or1k	z80
renesas	

Supported architectures on NuttX

What is POSIX?

The **Portable Operating System Interface** is a family of standards specified by the IEEE Computer Society for maintaining compatibility between operating systems. POSIX defines application programming interfaces (APIs), along with command line shells and utility interfaces, for software compatibility (portability) with variants of Unix and other operating systems. POSIX is also a trademark of the IEEE. POSIX is intended to be used by both application and system developers

- Source [POSIX - Wikipedia]

Advantage of POSIX

Instead of reinventing wheel again, you can reuse previously implemented a POSIX based application with NuttX.

Projects using NuttX

- PX4 autopilot drones.
- Pixhawk an advanced, User-Friendly Autopilot.
- OsmocomBB uses NuttX to develop an operating system for cell phones.
- Haltian's Thingsee IoT gateway devices use the ThingseeOS operating system, which is based on NuttX.
- Motorola Moto Z.
- Sony is using NuttX in their audio processors.
- Sony is using NuttX in the Spresense development board.
- Samsung TizenRT based on NuttX RTOS.
- Xiaomi Vela, an IoT software platform based on NuttX.

- Source NuttX - Wikipedia

Supported ESP Chips

RISC-V

- ESP32-C3
- ESP32-C6
- ESP32-H2

Xtensa

- ESP32
- ESP32-S2
- ESP32-S3

Peripheral	Support
ADC	No
AES	No
Bluetooth	Yes
CAN/TWAI	Yes
CDC Console	Yes
DMA	Yes
eFuse	No
GPIO	Yes
I2C	Yes
LED_PWM	Yes
RNG	No
RSA	No
RTC	Yes
SHA	No
SPI	Yes
SPIFLASH	Yes
Timers	Yes
Touch	No
UART	Yes
Watchdog	Yes
Wifi	Yes

ESP32-C3 Peripheral Support Status on NuttX

Peripheral	Support
ADC	YES
AES	YES
Bluetooth	No
CAMERA	No
CAN/TWAI	Yes
DMA	Yes
eFuse	No
GPIO	Yes
I2C	No
I2S	Yes
LCD	No
LED_PWM	No
MCPWM	Yes
Pulse_CNT	No
RMT	No
RNG	No
RSA	No
RTC	Yes
SD/MMC	Yes
SDIO	No
SHA	No
SPI	Yes
SPIFLASH	Yes
SPIRAM	Yes
Timers	Yes
Touch	Yes
UART	Yes
USB OTG	No
USB SERIAL	Yes
Watchdog	Yes
Wi-Fi	Yes

ESP32-S3 Peripheral Support Status on NuttX

Sample Application on NuttX

nuttx-apps/examples/hello/hello_main.c

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

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

Sample Application on NuttX

```
# Clean up the working directory
make -j distclean

# Setting up the example
./tools/configure.sh esp32s3-devkit:nsh

# Select the Hello World example
make menuconfig

# Compile and flash the firmware
make flash ESPTOOL_PORT=/dev/ttyUSB0 -j$(nproc)

# Access to terminal
minicom
```

Build steps

```
nsh> help
help usage:  help [-v] [<cmd>]

    .           cp          expr        ln          ps          true
    [           cmp         false       ls          pwd
truncate        dirname     fdinfo      mkdir       readlink    uname
    alias       dd          free        mkfifo      rm          umount
    unalias     df          help        mkrd        rmdir       unset
    arp         dmesg       hexdump     mount       set         uptime
    basename    echo        ifconfig    mv          sleep       usleep
    break       env         ifdown      nslookup    source      wget
    cat         exec        ifup        pidof       test        xd
    cd          exit        kill        printf      time

Builtin Apps:
    hello       nsh         ping        renew       sh          wapi
    iwasm       nxplayer    rand        rtpdump     stat        wget
nsh> hello
Hello, World!!
nsh>
```

Execute steps

Sample Application on NuttX

nuttx-apps/examples/gpio/gpio_main.c

```
/* Read the pin value */

ret = ioctl(fd, GPIOC_READ, (unsigned long)((uintptr_t)&invalue));
if (ret < 0)
  {
    int errcode = errno;
    fprintf(stderr, "ERROR: Failed to read value from %s: %d\n",
            devpath, errcode);
    close(fd);
    return EXIT_FAILURE;
  }
```

```
/* Open the pin driver */

fd = open(devpath, O_RDWR);
if (fd < 0)
  {
    int errcode = errno;
    fprintf(stderr, "ERROR: Failed to open %s: %d\n", devpath, errcode);
    return EXIT_FAILURE;
  }
```

```
/* Write the pin value */

ret = ioctl(fd, GPIOC_WRITE, (unsigned long)outvalue);
if (ret < 0)
 {
   int errcode = errno;
   fprintf(stderr,
           "ERROR: Failed to write value %u from %s: %d\n",
           (unsigned int)outvalue, devpath, errcode);
   close(fd);
   return EXIT_FAILURE;
 }
```

Brief History of ESP and NuttX

October 2016

First commit by Gregory Nutt himself (creator of NuttX), one month after the release of ESP32

October 2021

Espressif officially announces support for NuttX and Zephyr OSes (in addition to ESP-IDF, which remains Espressif's official OS for its SOCs)

November 2024

NuttX supports ESP32, ESP32-S2, ESP32-S3, ESP32-C3, ESP32-C6, and ESP32-H2

Next Steps

- WebAssembly
- elf-loader
- WiFi, Bluetooth, …

Articles and Updates About NuttX?

- NuttX · Developer Portal
- GitHub - apache/nuttx
- GitHub - apache/nuttx-apps
- Lup Yuen LEE's Blog

Questions? Thank You

Sources

- https://nuttx.apache.org/
- https://nuttx.apache.org/docs/latest/
- https://developer.espressif.com/blog/nuttx-getting-started/
- https://developer.espressif.com/blog/2024/11/using-wokwi-with-nuttx/
- https://developer.espressif.com/blog/nuttx-adding-porting-an-app/
- https://developer.espressif.com/blog/pytest-testing-with-nuttx/
- https://developer.espressif.com/tags/nuttx/
- https://en.wikipedia.org/wiki/Real-time_operating_system
- https://en.wikipedia.org/wiki/POSIX