

NuttX Real-Time Operating System

NuttX is a real-time operating system (RTOS) with an emphasis on standards compliance and small footprint. Scalable from 8-bit to 32-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()).

NuttX was first released in 2007 by [Gregory Nutt](#) under the permissive BSD license.

Key features

- Standards Compliant.
- Core Task Management.
- Modular design.
- Fully preemptible.
- Naturally scalable.
- Highly configurable.
- Easily extensible to new processor architectures, SoC architecture, or board architectures. See [Porting Guide](#).
- FIFO, round-robin, and “sporadic” scheduling.
- Realtime, deterministic, with support for priority inheritance.
- Tickless operation.
- POSIX/ANSI-like task controls, named message queues, counting semaphores, clocks/timers, signals, pthreads, robust mutexes, cancellation points, environment variables, filesystem.
- VxWorks-like task management and watchdog timers.
- BSD socket interface.
- Extensions to manage pre-emption.
- Optional tasks with address environments (*Processes*).
- Symmetric Multi-Processing (SMP)
- Loadable kernel modules; lightweight, embedded shared libraries.
- Memory Configurations: (1) Flat embedded build, (2) Protected build with MPU, and (3) Kernel build with MMU.
- Memory Allocators: (1) standard heap memory allocation, (2) granule allocator, (3) shared memory, and (4) dynamically sized, per-process heaps.
- Thread Local Storage (TLS)
- Inheritable “controlling terminals” and I/O redirection. Pseudo-terminals.
- On-demand paging.
- System logging
- May be built either as an open, flat embedded RTOS or as a separately built, secure *kernel* with a system call gate interface.
- Built-in, per-thread CPU load measurements.
- Custom NuttX C library
- Application interface well documented in the [NuttX User Guide](#).

Supported platforms

- ARM
 - ARM7TDMI (TI TMS320 C5471, Calypso, MoxART, NXP LPC214x, LPC2378, STMicro STR71x)
 - ARM920T (Freescale i.MX1)
 - ARM926EJS (TI DM320, NXP LPC31xx)
 - ARM Cortex-A5 (Atmel SAMA5D2, SAMA5D3, SAMA5D4)
 - ARM Cortex-A8 (Allwinner A10)
 - ARM Cortex-A9 (NXP/Freescale i.MX6)
 - ARM Cortex-R4/R4F (TI TMS570)
 - ARM Cortex-M0 (nuvoTon NUC120, NXP/Freescale KL25Z, KL26Z, Atmel SAMD20/21, SAML21, ST Micro STM32 F0)
 - ARM Cortex-M3 (ST Micro STM32 F1/L4/F2, TI/Stellaris LM3S, NXP LPC17xx, Atmel SAM3U/3X, SiliconLabs EFM32)
 - ARM Cortex-M4 (with/without floating point unit: ST Micro STM32 F3/F4/L4, TI/Stellaris LM4F/TM4C, NXP LPC43xx/LPC54xx, Freescale Kinetis K20/K40/60/64/66, Atmel SAM4C/4E/4S/4L, Infineon XMC4xxx)
 - ARM Cortex-M7 (Atmel SAMV71/SAME70, ST Micro STM32 F7)
- Atmel AVR
 - Atmel 8-bit AVR (AT90USB, ATmega)
 - AVR32
- Freescale M68HCS12
- Intel
 - 80x86
- MIPS
 - MicroChip PIC32MX (MIPS32 24Kc)
 - MicroChip PIC32MZ (MIPS32 M14k)
- Misoc
 - LM32 (Qemu)
- Renesas/Hitachi
 - Renesas/Hitachi SuperH
 - Renesas M16C/26
- RISC-V
- Xtensa LX6
 - Expressif ESP32
- Zilog
 - Zilog Z16F ZNeo
 - Zilog eZ80 Acclaim!
 - Zilog Z8Encore!
 - Zilog Z80
- Complete List

File system

- Tiny in-memory, root pseudo-file-system.
- Virtual File System (VFS)
- Mount-able volumes. Bind mountpoint, filesystem, and block device driver.
- Generic system logging (SYSLOG) support.
- FAT12/16/32 filesystem support.
- NFS Client. Client side support for a Network File System (NFS, version 3, UDP).
- NXFFS. The tiny NuttX wear-leveling FLASH file system.

- SMART. FLASH file system from Ken Pettit.
- ROMFS file system support.
- BINFS pseudo-file system support.
- HOSTFS file system support (simulation only).
- Union file system. Supports combining and overlaying file systems
- UserFS. Provides a user application file system.
- procfs/ pseudo-file system.
- Generic driver for SPI-based MMC/SD/SDH cards.
- A Binary Loader with support for the following formats:
 - Separately linked ELF modules.
 - Separately linked `NXFLAT` modules. `NXFLAT` is a binary format that can be XIP from a file system.
- PATH variable support.
- File transfers via TFTP and FTP (get and put), HTML (wget), and Zmodem (sz and rz).
- Intel HEX conversions.

Device Drivers

- VFS supports character and block drivers.
- Asynchronous I/O (AIO).
- Network, USB (host), USB (device), serial, I2C, I2S, NAND, CAN, ADC, DAC, PWM, Quadrature Encoder, generic timer, and watchdog timer driver architectures.
- RAMDISK, pipes, FIFO, /dev/null, /dev/zero, /dev/random, and loop drivers.
- Generic driver for SPI-based or SDIO-based MMC/SD/SDH cards.
- Power Management sub-system.
- ModBus support provided by built-in FreeModBus version 1.5.0.
- Graphics Devices: framebuffer drivers, graphic and segment LCD drivers
- Audio Subsystem: CODECs, audio input and output drivers. Command line and graphic media player applications.
- Cryptographic sub-system.
- Input Devices: Touchscreen, USB keyboard, USB mouse, analog/discrete joystick, GPIO-based buttons and keypads.
- Memory Technology Devices
- Analog Devices: Support for Analog-to-Digital conversion (ADC), Digital-to-Analog conversion (DAC), multiplexers, and amplifiers.
- System Logging Devices.

C/C++ Library

- Standard C Library Fully integrated into the OS.
- Includes floating point support via a Standard Math Library.
- Add-on uClibc++ module provides Standard C++ Library supporting iostreams, strings, STL, RTTI, exceptions, etc. (LGPL).
- Contemporary port of the C++11 *LLVM libc++* is also available.

Networking

- Multiple network interface support; multiple network link layer support
- IPv4, IPv6, TCP/IP, UDP, ARP, ICMP, ICMPv6, IGMPv2 (client) stacks.
- User space stacks
- Stream, datagram, and raw packet sockets
- Address Families: IPv4/IPv6 (*AF_INET,AF_INET6*), Raw socket (*AF_PACKET*), raw IEEE 802.15.4 (*AF_IEEE802154*), and local, Unix domain socket support (*AF_LOCAL*).
- Special *INET* protocol sockets: Raw ICMP and ICMPv6 protocol ping sockets (*IPPROTO_ICMP, IPPROTO_ICMP6*).
- Custom user sockets.
- IP forwarding
- DNS name resolution / NetDB
- IEEE 802.11 FullMac
- Radio Network Drivers: IEEE 802.15.4 MAC, Generic Packet Radio
- 6LoWPAN for radio network drivers (IEEE 802.15.4 MAC and generic packet radios)
- SLIP, TUN/PPP, local loopback devices
- A cJSON port
- Small footprint.
- BSD compatible socket layer.
- Networking utilities (DHCP server and client, SMTP client, TELNET server and client, FTP server and client, TFTP client, HTTP server and client, PPPD, NTP client). Inheritable TELNET server sessions (as “controlling terminal”). VNC server.
- ICMPv6 autonomous auto-configuration
- NFS Client. Client side support for a Network File System (NFS, version 3, UDP).
- A NuttX port of Jeff Poskanzer's THTTPD HTTP server integrated with [NXFLAT](#) to provide embedded CGI.
- PHY Link Status Management
- UDP Network Discvory, XML RPC Server.
- XML RPC Server
- Support for network modules (such as the TI CC3000 WLAN module and ESP8266).

Flash Support

- MTD-inspired interface for Memory Technology Devices.
- NAND Support.
- FTL. Simple Flash Translation Layer support file systems on FLASH.
- NXFFS. the NuttX wear-leveling FLASH file system.
- Support for SPI-based FLASH devices.

USB Support

USB Host Support

- USB host architecture for USB host controller drivers and device-dependent USB class drivers.
- USB host controller drivers available for the Atmel SAMA5Dx, NXP LPC17xx, LPC31xx, and STmicro STM32.

- Device-dependent USB class drivers available for USB mass storage, CDC/ACM serial, HID keyboard, HID mouse, and Xbox game controller.
- Seam-less support for USB hubs.

USB Device Support

- *Gadget*-like architecture for USB device controller drivers and device-dependent USB class drivers.
- USB device controller drivers available for the PIC32, Atmel AVR, SAM3, SAM4, and SAMA5Dx, NXP LPC17xx, LPC214x, LPC313x, and LPC43xx, Silicon Laboratories EFM32, STMicro STM32 F1, F2, F3, and F4, and TI DM320, and others.
- Device-dependent USB class drivers available for USB mass storage, CDC/ACM serial, RNDIS networking, and PL2303 serial emulation.
- Dynamically configurable, composite USB devices.
- Built-in USB trace functionality for USB debug.

Graphics Support

- Framebuffer drivers.
- Graphic LCD drivers for both parallel and SPI LCDs and OLEDs.
- Segment LCD drivers
- VNC Server
- `mmap`-able, framebuffer character driver
- NX Graphics Subsystem: A graphics library, windowing system and font support that works with either framebuffer or LCD drivers.
- NuttX Widgets: A graphical user interface written in conservative C++ that integrates with NX Graphics.
- NuttX Window Manager: A tiny window manager based on the NX Graphics Subsystem and NuttX Widgets.

Add-Ons

NuttShell

A small, scalable, bash-like shell for NuttX with rich feature set and small footprint. See the [NuttShell User Guide](#).

Pascal Runtime

The Pascal add-on is available for download from the [Bitbucket.org](http://www.bitbucket.org) website.

apps/ Package

The **apps/** package is not strictly a part of NuttX but is retained in the NuttX GIT repository and is released as a separate versioned package. The **apps/** package contains a collection of NuttX applications including the NuttShell (NSH), networking utilities, tests and examples, interpreters, and NSH extension commands.

From:
<http://www.nuttx.org/> - **NuttX Real-Time Operating System**

Permanent link:
<http://www.nuttx.org/doku.php?id=nuttx>

Last update: **2018/03/01 18:04**

