

LPCOpen releases and versioning information

AS OF V2.04, THIS CONSOLIATED VERSION HISTORY PAGE IS NO LONGER SUPPORTED. IT has been replaced with device specific version history pages that are linked to the LPCOpen download pages. You can find out the version history for a specific release by selecting the HISTORY link next to the download links on the device download pages.



LPCOpen release history

Older versions of the LPCOpen can be download on the [LPCOpen download page](#).

LPCOpen version changes and known issues

Some issues are known at the time of the versioned package release. Issues found after the release can be found on the LPCOpen bug tracker pages.

LPCOpen v2.xx release history

LPCOpen v2.03 release (Released: 11/22/2013): LPC11U37 release only

The v2.03 release adds support for the LPCXpresso LPC11U37 board.

Changes

- None - initial release.

Known issues

- The LPC11xx documentation package does not include this board yet and has not been updated.

LPCOpen v2.02 release (Released: 11/12/2013): LPC18xx/43xx updates only

The v2.02 update adds support for the Hitex LPC1857 and LPC4357 boards, the Keil MCB1800 and MCB4300 boards, the NGX Xplorer LPC1830 and LPC4330 boards, and the LPC-Link 2 (LPC4370) board.

Changes

- All projects converted to v2.xx format - separate projects for IAR and Keil 'per board' and LPCXpresso archived workspace projects.
- Chip layer code and Board layer code are no longer blended into a single platform library.
- The lpc_ip layer has been completely removed and all functionality has been moved the chip layer. Files in this area had to be selectively added to projects 'per device' along with chip layer code - adding is no longer needed. Removing this layer has also allow the chip layer code to be a bit smaller. In almost all cases, the Chip layer APIs were not altered with this change.
- FreeRTOS source code updated from v7.3.0 to V7.4.
- LPC18xx/43xx chip layer changes
 - A new HSADC driver and example are included for the LPC4370 devices and the LPC-Link 2 board
 - Combined GPIO driver separated into 3 drivers: GPIO, PININT, and GPIO group interrupt
 - All chip layer code for a specific device family can now be (mass) added to a single project and only the files specific to the device in that family will built
 - The 'sys_config.h' file is now a chip file and only details which CHIP_* definition to use when building the code. Pre-populated sys_config.h files are available for all supported devices in a single family in the chip layer area. Just add the include path for your device to your project to build for that specific device
 - A chip specific Chip_SystemInit() is now provided that sets up the system to boot using the IRC/PLL, but doesn't setup pin muxing or memory
 - Moved SystemCoreClockUpdate() function and SystemCoreClock variable from board the chip layer
 - Chip layer code no longer requires definitions that were defined in the board layer such as oscillator rate, internal clock input rate, etc.
 - For these few cases where these were used, the chip layer now uses them as external constants defined somewhere else (usually in the board layer)

LPCOpen releases and versioning information

- Very minor API changes for some drivers
- LPC18xx/43xx board layer changes
 - Shared 18xx and 43xx board layer code is now separate to make changed and packaging easier
 - Named board header file merged with board.h (this extra named file wasn't needed)
 - DEBUG_* definitions used to control DEBUG_* functions are now part of the board.h file
 - Named board source and system init files renamed to generic named files board.c and board_sysinit.c (allows simply overwriting the files to change the board)
 - SystemInit() function renamed to Board_SystemInit()
- Projects
 - Keil and IAR project cleanup to improve consistency
 - LPCXpresso projects are now contained in archives and are much simpler to use
 - Projects are no longer shared across platforms, projects are 1 per board
 - The file containing SystemInit() is included in each project example now

Known issues

- The LPCXpresso versions of the webserver application fail to load the web page after the first load
- The API documentation package is not yet released with this update
- Dual-core examples are not yet available in the LPCXpresso projects for LPC43xx based boards
- periph_atimer example doesn't work correctly. It only fires once.

LPCOpen v2.01 release (Released: 10/4/2013): LPC8xx updates only

The v2.01 update adds support for the LPC800_MAX board and some small tweaks in the LPC8xx chip layer code.

Changes

- Added support for LPC800-MAX board with LPC812 device for Keil, IAR, and LPCXpresso toolchains.
- LPC8xx chip layer changes
 - SCT driver register structured reorganized to include 32x1/16x2 unions. Should not impact existing code.
 - For fixed pin enumerations in the swm_8xx.h file, 'SWM_FIXED_' now precedes the definition name to identify it was a SWM fixed pin definition: Example: ACMP_I1 changed to SWM_FIXED_ACMP_I1
 - Minor commenting and formatting fixes, some DoxyGen related cleanup
 - Fixed logic for Chip_SWM_IsEnabled() function
 - Header files for ROM functions are now prefixed with rom_ (example rom_uart_8xx.h)
 - Moved some functions related to clocking from syscon_8xx.h to clock_8xx.h
 - Added new function Chip_UART_GetIntsEnabled() for returning enabled UART interrupts
 - CMSIS IRQ names (NVIC) mapped to interrupt handler function names in documentation now
 - cmsis.h file was moved inside the chip_8xx area
 - Fixed offset for DEVICEID for LPC8xx in SYSCON (submitted by developer)
 - Renamed Chip_PININT_IsHighEnable() to Vhip_PININT_GetHighEnabled() and Chip_PININT_IsLowEnabled to Chip_PININT_GetLowEnabled()
 - 'PMI' convention changed to 'PININT' convention for PININT driver, LPC_PMI changed to LPC_PININT
 - Added build warnings to some examples when pin muxing hasn't been defined
 - Keil and IAR project cleanup to improve consistency
 - Added a 'board_api_stubs_8xx.c' file to use as a drop-in board layer stub

Known issues

- periph_i2c_rom_slave example builds and runs, but does not work correctly. (LPC8xx release only)
- ACMP ladder selection is wrong.

LPCOpen v2.00a release (Released: 09/13/2013)

The v2.00a update adds updated LPC11xx support and the initial driver API and example documentation releases for the LPC8xx and the LPC11xx.

Changes

- Major changes have been made to the LPC11xx package and structure that are similar or the same as the LPC8xx changes. See the LPC8xx v2.00 changes (except those specific to the LPC8xx) for a full list of these similar changes.
- Initial release of USBBD library and examples - The USBBD library provides a USB API that is the same as the USB ROM API, but works with a static library instead. This allows examples that would normally only work with devices that have the USB ROM API to work on devices without the USB ROM API with little or no changes.
 - The Windows USB driver package needed for the USBBD library examples (most examples don't need them) is download as a separate package. The Windows USB driver package has a different versioning scheme than LPCOpen and is released on a different cycle.
- LPC8xx only
 - Release of the initial LPC8xx documentation for the driver API and example code
- LPC11xx only changes
 - Added new USBBD library support and USBBD library examples (provides same API as USB ROM API based devices in library form). Also includes a composite example.
 - New PININT driver, new GPIO group driver, new PMU driver
 - Added additional examples for PININT/GPIOINT. Improved some existing examples
 - Added initial support for the LPC1125, new LPC1125 sequenced ADC driver and example
 - Fixed Chip_TIMER_ExtMatchControlSet() function per bug tracker issue
 - Standardized API for GPIO drivers with added port and better optimized functions
 - Removed GPIO_Chip_FIO_* functions - not applicable to 11xx devices
 - UART driver improvements to help with non-blocking and interrupt support
 - Fixed a potential issue with startup code related to PLL setup while it was powered up
 - ROM API standardization and alignment with other device families
 - Driver cleanup and removal of conditional code not related to the LPC11xx device family
 - Release of the initial LPC11xx documentation for the driver API and example code

Known issues

- The LPC11xx documentation package shows some drivers sections with no information on the driver API. This is related to how DoxyGen parses the #ifdef's embedded in the code and will be fixed in a future release. Please see the API comments in the header files until this is fixed.

LPCOpen v2.00 release for LPC8xx only (Released: 08/30/2013)

Changes

- LPCOpen changes (also applies to upcoming LPC11xx and LPC13xx releases, but not LPC17xx/40xx and LPC18xx/43xx releases)
 - Chip layer code and Board layer code are no longer blended into a single platform library
 - Chip layer code no longer requires definitions that were defined in the board layer such as oscillator rate, internal clock input rate, etc.
 - For these few cases where these were used, the chip layer now uses them as external constants defined somewhere else (usually in the board layer)
 - The lpc_ip layer has been completely removed and all functionality has been moved the chip layer. Files in this area had to be selectively added to projects 'per device' along with chip layer code - adding is no longer needed. Removing this layer has also allow the chip layer code to be a bit smaller. In almost all cases, the Chip layer APIs were not altered with this change.
 - Chip layer changes (./lpcopen/lpc_core/lpc_chip/<chip>)
 - All chip layer code for a specific device family can now be (mass) added to a single project and only the files specific to the device in that family will build

- The 'sys_config.h" file is now a chip file and only details which CHIP_* definition to use when building the code. Pre-populated sys_config.h files are available for all supported devices in a single family in the chip layer area. Just add the include path for your device to your project to build for that specific device
- A chip specific Chip_SystemInit() is now provided that sets up the system to boot using the IRC/PLL, but doesn't setup pin muxing or memory
- Moved SystemCoreClockUpdate() function and SystemCoreClock variable from board to the chip layer
- Board layer changes (./lpcopen/lpc_core/lpc_board/<chip>/<board>)
 - Named board header file merged with board.h (this extra named file wasn't needed)
 - DEBUG_* definitions used to control DEBUG_* functions are now part of the board.h file
 - Named board source and system init files renamed to generic named files board.c and board_sysinit.c (allows simply overwriting the files to change the board)
 - SystemInit() function renamed to Board_SystemInit()
- FreeRTOS source code updated from v7.3.0 to V7.4.2
- Package release model changed from flat/combined release to include packaged Keil/IAR release and dedicated LPCXpresso archive
- Very minor API changes for some drivers
- Keil and IAR projects are relocated into a central area and are 'per board' instead of 'per device'
 - This prevents projects that have support for many boards and selection of the wrong target board
 - Allows easier packaging of LPCOpen 'per board' and easier long-term maintenance of the projects
- LPCXpresso projects have been replaced by LPCXpresso project XML import files
 - The import files can be used to generate the LPCXpresso projects in the LPCOpen flat release by simply importing them
- The file containing SystemInit() is included in each project example now
- LPC8xx only changes
 - Major rewrite of some LPC8xx drivers
 - New ring buffer based UART driver, improved GPIO functions, new PININT driver
 - Improvements and new functions for FMC, PMU, SYSCON, IOCON, WKT, MRT, CRC, WDT, ROM APIs
 - Uses enumeration for indexing IOCON registers: Example: LPC_IOCON->PIO0[IOCON_PIO0] = 0x1; // LPC_IOCON->PIO0[0] is not correct!
 - Added additional examples for SCT, UART, GPIO/PINIT, ROM APIs, etc. Improved some existing examples
 - Some projects support CRP and MTB
 - SystemInit() code has the option of using the ROM PLL setup functions instead of the clock driver

Known issues

- periph_i2c_rom_slave example builds and runs, but does not work correctly.

LPCOpen v1.xx release history

LPCOpen v1.03 release (Released: 05/10/2013)

Changes

- Updated I2C driver and examples for all platforms (much smaller and easier to use), cleaner APIs for master and slave support
- Fully tested support for the LPC175x/6x device family
- Board and example support for the Xpresso LPC1769 board
- LPC17xx/40xx uCos-III multiple task example
- Documentation: Porting LPCOpen to a new board or platform
- Documentation: Creating project files with LPCOpen

LPCOpen releases and versioning information

- Better documentation for build-time definitions (sys_config.h)
- List of required IP files per platform
- List of required include paths needed per platform (IP, Chip, and Board layers)
- Change the MD_* definitions (ie, MD_PUP) to names that are meaningful for IOCON functions
- SDMMC driver optimization - break enumeration sequence from the driver, simplify and optimize driver
- Other small optimizations to reduce memory footprint
- SD card and LWIP benchmarks plus related examples
- Additional code and examples: 18xx/43xx EEPROM support, more LWIP (webserver) examples, NOR and NAND FLASH examples for 17xx/40xx, and more.
- Updates and fixes to the LPC1343 code base (no platform yet)

Known issues

- Timer clock selection on 11xx (and possibly others) is incorrect.
- LPC18xx/43xx SCU mode pin macro values are not correct. This may impact some examples that require the correct pullup/pulldown/repeater state on those pins.
- **LPCOpen v1.03 currently supports the v5.18 emWin libraries and not the v5.20 libraries (yet).**
- **For the LPC13xx examples, example build targets exist for the Xpresso LPC1343 board in the Keil and IAR projects, but will build with errors since the Xpresso LPC1343 Board layer isn't included in this release. Full Xpresso LPC1343 board support will be in the next release.**
- Time delay functions are not accurate and are currently CPU cycle based. Some for-loop operations are being optimized out by the compiler and need to be replaced with time delay calls. These optimized out delays are rare and don't appear to be causing any issues.
- LPC4088 LWIP examples for IAR toolchain may not work properly - image can be FLASHed, but not always debuggable, can usually run standalone. (Reason unknown)
- Support for the following device families is available in the platform, but is not fully tested. LPC1343, LPC11xx, LPC11LVxx, LPC11Dxx, and LPC11Exx. The LPC1343 Chip layer should be fully operational, but does not have working examples yet.
- Some UART drivers shared a common buffer, so using multiple UART devices simultaneously will corrupt buffer data. This doesn't apply to the LPC11xx and LPC13xx drivers.

LPCOpen v1.02 release (Released: 03/21/2013)

Changes

- IAR EWARM projects for all supported LPCOpen boards (requires EWARM v6.50.2 or later)
- Xpresso support for 17xx/40xx boards (requires Xpresso 5 v5.1.2_2065 or later)
- LWIP support for 17xx/40xx added
- Xpresso projects for USBLIB have been moved under the applications area for 11U14, EA1788, EA4088, and LPC1347 boards (makes imports easier)
- Several new examples for some devices
- Additional 18xx/43xx Xpresso project cleanup plus missing projects
- Improved quick start guides for all toolchains
- Some small API changes to help reduce memory footprint

Known issues

- Time delay functions are not accurate and are currently CPU cycle based. Some for-loop operations are being optimized out by the compiler and need to be replaced with time delay calls. These optimized out delays are rare and don't appear to be causing any issues.
- For Keil MCB18xx/43xx boards, the NOR FLASH setup has not been fully tested. It isn't currently used in any examples
- Startup code for the 18xx/43xx does not setup the external bus signals, so it should be added by the end-user if needed for a board design

LPCOpen releases and versioning information

- LPC4088 LWIP examples for IAR toolchain may not work properly - image can be FLASHed, but not always debuggable, can usually run standalone.
- Support for the following device families is available in the platform, but is not fully tested. LPC1343, LPC175x/6x, LPC11xx, LPC11LVxx, LPC11Dxx, and LPC11Exx
- Some UART and I2C drivers shared a common buffer, so using multiple UART or I2C devices simultaneously will corrupt buffer data (will fix in v1.03)

LPCOpen v1.01 release (Released: 03/01/2013)

Changes

- Restores LPCXpresso-IDE projects for the 18xx/43xx platforms, LPCXpresso-IDE projects are now centered around boards instead of devices and should be much easier to use
 - Revised (and much simpler) documentation for LPC18xx/43xx LPCXpresso-IDE examples
- Adds LPC13xx CHIP support - this includes the LPC1343 and LPC1347 device families
- Adds LPCXpresso LPC1347 board support with tool chain support for Keil Uvision4, IAR Embedded Workbench, and version 5 of the LPCXpresso-IDE
- Adds LPCXpresso LPC11C24 board support with tool chain support for Keil Uvision4 and version 5 of the LPCXpresso-IDE
- Adds LPCXpresso LPC11U14 board support with tool chain support for Keil Uvision4 (support for version 5 of the LPCXpresso-IDE is already available in v1.00)
- Introduces a new UART driver model (uart_004) which should be much smaller and more efficient than previous UART drivers, currently used for the LPC11xx and LPC13xx platforms only
- Lots of fixes related to code standardization - commenting, APIs, types, etc.
- Various tweaks to reduce final image size - API changes, code optimization, etc.
- Support for uC/OS-III including LCP18xx/43xx projects using uC/OS-III

Known issues

- IAR projects didn't make it into v1.01, but should be available in v1.02
- Paths used for projects can get very large and cause environment issues, read the installation documentation for information on working around this issue
- Time delay functions are not accurate and are currently CPU cycle based
- For Keil MCB18xx/43xx boards, the NOR FLASH setup has not been fully tested. It isn't currently used in any examples
- Startup code for the 18xx/43xx does not setup the external bus signals, so it should be added by the end-user if needed for a board design
- Although most LPCXpresso-IDE projects are available for the LPC18xx/43xx boards, some boards don't have all the projects available yet

LPCOpen v1.00 release

Changes

- More LPC43xx dual-core and misc. (i.e., OTP, SCT, etc.) examples
- Added LPC11xx platform support with initial support for the LPCXpresso LPC11U14 board with version 5 of the LPCXpresso-IDE
- Added LPX8xx platform support with initial support for the LPCXpresso LPC812 board with version 5 of the LPCXpresso-IDE and Keil UV4
- Added LPC17xx/40xx platform support with initial support for the Embedded Artist's LPC1788 and LPC4088 boards board with Keil UV4
- Lots of low-level optimization of drivers
- Added support for Chan FATFS file system
- Lots of code documentation improvements
- Updated many source and header files to standard look and feel

Known issues

- LPC18xx/43xx LPCXpresso-IDE and IAR projects have been temporarily removed from the platform, these will be re-enabled in v1.01
- Paths used for projects can get very large and cause environment issues, read the installation documentation for information on working around this issue
- USB host examples for the Keil 1857/4357 boards require external power to work. This isn't explained in the documentation.
- When adding a USB to UART convertor to the LPC8xx board, center the analog pot to reduce loading on the UART RX signal or receive won't work. It is also highly recommended to also remove the wires for CTS and RTS.

LPCOpen v0.51 release

Changes

- Complete restructuring of the LPCOpen platform
- Added support for the Hitex 4350, Keil 4357, and NGX Xplorer 4330 boards. Updated and standardized Keil uVision4 and LPCXpresso-IDE version 4 toolchain support. IAR EWARM support added.
- Expanded LPCOpen/example documentation and added online build pages
- Examples and driver APIs refined to remove some abstraction, examples are specific to architecture now
- Early system init code standardized among boards
- Lots of small driver API and structure tweaks to help with speed and size
- Lots of code and comment cleanup, coding standards and style has been standardized
- Added SWIM code to software area and SWIM example in the 18xx/43xx examples area
- IP, CHIP, and BOARD libs are no longer separate - a single board library file contains all the needed support for a single board
- LWIP and FreeRTOS are no longer built as libraries
- Added multiple configurable dual-core examples

Known issues

- Some 18xx IAR projects may not work correctly with all LPC18xx boards (this might be related to the IAR toolchain version) (will fix in v0.90)
- All 18xx/43xx board variants for a specific example/project may not be available (will fix in v0.90) - for example, a project may be available for Keil UV4, but not yet Xpresso4
- 18xx/43xx clock driver returns a rate of 0 for audio PLL or USB PLL, avoid using this function for those 2 clocks
- For Keil MCB18xx/43xx boards, the NOR FLASH setup has not been fully tested. It isn't currently used in any examples.
- Board button support functions don't work correctly on NGX18xx/43xx and Hitex18xx/43xx boards (will fix in v0.90)
- 18xx/43xx Board_LED_Test() and Board_LED_Toggle() functions do not work correctly (will fix in v0.90)
- 18xx/43xx Time delay functions are not accurate and are currently CPU cycle based (will fix in v0.90)
- Some projects may give build warnings due to unused variables, etc. in some build configurations
- Keep the base directory for the LPCOpen directory close to your root. Keil and IAR tools have issues with long path names! If you are seeing build errors related to not finding files (i.e., header files missing), this may be the issue.

LPCOpen ALPHA 1 release

Changes

- Initial release of the LPCOpen platform with limited support for the Hitex 1850, Keil 1857, and NGX Xplorer 1830 boards. Supports Keil uVision4 and LPCXpresso-IDE version 4.

Known issues

- example_ssp: Spurious verify fail message when programming to internal flash with Keil
- example_emWin_with_touch: Touch analog inputs update slowly
- example_emWin_with_touch: Numbers update more slowly in the LPCXpresso-IDE
- example_emc: RAM size for flash algorithm in Keil I_Flash too small
- example_wdt: Missing flash algorithm setup for all Keil configurations
- example_blinky_iponly: Keil Build output folders incorrect
- All Examples : 1857 internal flash programming support in LPCXpresso-IDE not released yet
- Documentation Modules Index : Needs organization- Not all the groups appear in the right spot in the tree
- example_rtc: Does not use udl_board layer to control LEDs
- Everything: DoxyGen documentation missing or not detailed enough
- example_fir: Need example showing use of bundled CMSIS DSP library
- example_spifi: Need example showing use of SPIFI library
- All Examples: Extra untested build targets. Tested targets are: Keil I_Flash, Keil E_QSPIFlash, Xplorer E_QSPIFlash
- Everything: Warnings emitted during compiles
- Hardware Instructions : Xplorer board does not boot when attached to LPCXpresso base board
- LPCXpresso-IDE Projects : LPCXpresso-IDE projects run in debugger but do not boot without semihost modifications

How to Reach Us

Home Page: www.nxp.com

Web Support: www.nxp.com/support

USA/Europe or Locations Not Listed:

NXP Semiconductor

Technical Information Center, EL516

2100 East Elliot Road

Tempe, Arizona 85284

+1-800-521-6274 or +1-480-768-2130

www.nxp.com/support

Europe, Middle East, and Africa:

NXP Halbleiter Deutschland GmbH

Technical Information Center

Schatzbogen 7

81829 Muenchen, Germany

+44 1296 380 456 (English)

+46 8 52200080 (English)

+49 89 92103 559 (German)

+33 1 69 35 48 48 (French)

www.nxp.com/support

Japan:

NXP Semiconductor

ARCO Tower 15F

1-8-1, Shimo-Meguro, Meguro-ku,

Tokyo 153-0064, Japan

0120 191014 or +81 3 5437 9125

support.japan@nxp.com

Asia/Pacific:

NXP Semiconductor Hong Kong Ltd.

Technical Information Center

2 Dai King Street

Tai Po Industrial Estate

Tai Po, N.T., Hong Kong

+800 2666 8080

support.asia@nxp.com
