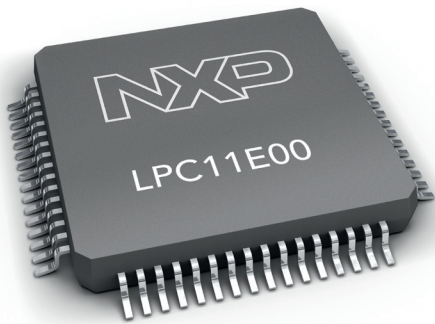




NXP 50-MHz, 32-bit ARM Cortex-M0™ microcontrollers LPC11E00

Feature-rich Cortex-M0 EEPROM solutions with USART, I²C & two SSP

Built around the Cortex-M0 architecture – the smallest, lowest power, and most energy-efficient ARM core ever developed – the LPC11E00 series is ideal for use in low-power 8/16-bit applications needing a feature-rich EEPROM solution. Also included are on-chip ROM drivers for the EEPROM, 32-bit divide, and power management.

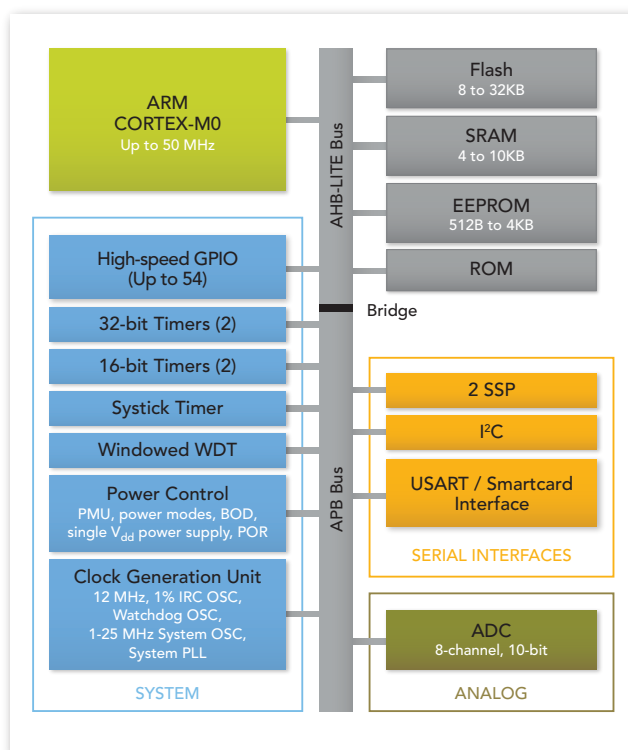
Features

- ▶ ARM Cortex-M0 processor, running at frequencies of up to 50 MHz
- ▶ Memory:
 - Up to 32 kB on-chip flash program memory
 - Total of 10 kB SRAM data memory
 - Up to 4 kB EEPROM
 - EEPROM API, 32-bit Divide API
 - In-System Programming (ISP-UART) and In-Application Programming (IAP) via on-chip bootloader software
- ▶ Debug options:
 - Standard JTAG test/debug interface. Serial Wire Debug
 - Boundary scan for simplified board testing
- ▶ Up to 54 General-Purpose I/O (GPIO) pins with configurable pull-up/pull-down resistors, repeater mode, and open-drain mode
 - Two GPIO grouped interrupt modules enable an interrupt based on a programmable pattern of input states of a group of GPIO pins
- High-current source output driver (20 mA) on one pin (P0_7)
- High-current sink driver (20 mA) on two true open-drain pins (P0_4 and P0_5)
- ▶ Four general-purpose counter/timers with a total of up to 5 capture inputs and 13 match outputs
- ▶ Programmable Windowed WatchDog Timer (WWDT) with a dedicated, internal low-power WatchDog Oscillator (WDO)
- ▶ Analog peripherals:
 - 10-bit ADC with input multiplexing among eight pins
- ▶ Serial interfaces:
 - USART with fractional baud-rate generation
 - USART supports an asynchronous Smart Card interface (ISO 7816-3)
 - Two SSP controllers with FIFO and multi-protocol capabilities
 - I²C-bus interface supporting the full I²C-bus specification and Fast-mode Plus (Fm+)



- ▶ Clock generation:
 - Crystal oscillator with an operating range of 1 to 25 MHz (system oscillator)
 - 12 MHz Internal RC oscillator (IRC) trimmed to 1% accuracy
 - Internal low-power, low-frequency WatchDog Oscillator
 - PLL allows CPU operation up to the maximum CPU rate with the system oscillator or the IRC as clock sources
 - Clock output function
- ▶ Power control:
 - Four reduced power modes: Sleep, Deep-sleep, Power-down, and Deep power-down
 - Power profiles residing in boot ROM
 - Processor wake-up from Deep-sleep and Power-down modes via reset, selectable GPIO pins, watchdog interrupt
- ▶ Power-On Reset (POR)
- ▶ Brownout detect with four separate thresholds for interrupt and forced reset
- ▶ Unique device serial number for identification
- ▶ Single 3.3 V power supply (1.8 to 3.6 V)
- ▶ Temperature range -40 to +85 °C
- ▶ Available in 48-pin LQFP, 64-pin LQFP, and 33-pin HVQFN package

LPC11E00



Maximizing connectivity, minimizing power

Connectivity options on the LPC11E00 series include two Synchronous Serial Port (SSP) interfaces, I²C with Fast-mode Plus feature for 10x higher bus-drive capability, a Universal Synchronous-Asynchronous Receiver/Transmitter (USART) and a Smart Card interface. The Smart Card interface (ISO7816-3) provides a plug-and-play interface for Smart Cards, making the LPC11E00 a good fit for e-commerce applications.

Development tools

NXP's easy-to-use LPCXpresso IDE is priced under \$30. Other development tools, including options from IAR, Keil, Hitex, and Code Red, can be found at www.nxp.com/microcontrollers. All of NXP's Cortex-M microcontrollers are upwardly binary compatible and offer the advantages of a single development toolchain. Developers can easily migrate their designs between Cortex-M0 and Cortex-M3 with minimal effort.

LPC11E00 selection guide

Type number	Flash	RAM	UART	EEPROM	I ² C	Standard timers	PWM channels	SSP	ADC channels	Package
LPC11E11	8	4 KB	1	512 B	1	5	11*	2	8/ 10b	HVQFN33
LPC11E12	16	6 KB	1	1 KB	1	5	11*	2	8/ 10b	HVQFN33 (5x5), TFBGA48
LPC11E13	24	8 KB	1	2 KB	1	5	11*	2	8/ 10b	LQFP64, LQFP48, HVQFN33
LPC11E14	32	10 KB	1	4 KB	1	5	11*	2	8/ 10b	LQFP64, LQFP48, HVQFN33

* Using timers