



The New ARM Cortex-M0 Processor

Meeting the Demands
of Low Power Applications

Low Power Design using the LPC1100 Series, A Methodology for
Low Power Verification, Power Management for Optimal Power Design

PLUS: Techniques for Optimization of Audio Codecs
Providing a Clear Vision for Home Entertainment, Printer with an Attitude and MORE!

ARM Cortex-M0 Processor

*Meeting the demands
of tomorrow's low
power applications*



T This special section introduces the ARM Cortex-M0, the smallest, lowest-power, and most energy-efficient ARM processor available. The exceptionally small silicon area, low-power and minimal code footprint of the processor enables developers to achieve 32-bit performance at an 8-bit price point.

The articles in this section include:

ARM Cortex-M0 Processor Introduction By Dominic Pajak, ARM

The NXP LPC1100 ARM Cortex-M0 MCUs By Rob Cosaro, NXP

Low Power Design using the LPC1100 Series By Rob Cosaro, NXP

The new Cortex-M0 Processor further extends ARM's MCU roadmap into ultra low-power MCU and SoC applications including:

- Gaming accessories
- Lighting
- Motor Control
- e-Metering
- Smart control
- Analog and Mixed Signal
- Power Control
- Medical Devices
- Zigbee & Z-Wave Systems

Introduction to the ARM Cortex-M0

By Dominic Pajak, ARM

As the cost of energy continues to grow, and concern about the environment matures, the increasing penetration of embedded devices into everyday lives presents developers with the challenge of managing the trade off between the demands for performance and low-power. Traditionally 16-bit microcontrollers have been used to provide the low-power consumption required, but in today's applications their lack of performance efficiency can mean shorter battery life. To meet this challenge ARM has developed a processor combining the performance of 32-bit, with the lower power and gate count normally associated with 16-bit processors.

Introduction The demand for ever lower-cost products with increasing connectivity (e.g. USB, Bluetooth, IEEE 802.15) and sophisticated analog sensors (e.g. accelerometers, touch screens) has resulted in the need to more tightly integrate analog devices with digital functionality to pre-process and communicate data.

Most 8-bit devices do not offer the performance to sustain these tasks without significant increases in MHz and therefore power, and so embedded developers are required to look for alternative devices with more advanced processor technology. The 16-bit devices have previously been used to address energy efficiency concerns in microcontroller applications. However, the relative performance inefficiencies of 16-bit devices mean they will generally require a longer active duty cycle or higher clock frequency to accomplish the same task as a 32-bit device.

The 32-bit ARM Cortex-M0 processor has been developed to address this need for increased performance efficiency while remaining very low-power, making it ideal for the next generation of ultra low-power MCUs and precision analog devices.

ARM Cortex-M0 Processor. The ARM Cortex-M processor family is specifically designed to address the needs of deeply embedded applications that require low-power and fast interrupt response, making it ideal for microcontrollers. The flagship processor in this range today is the ARM Cortex-M3, offering superior performance and features. The new ARM Cortex-M0 complements this by enabling silicon vendors to offer devices with an upwards compatible subset of the ARM Cortex-M3 features at an even lower area and power.

The ARM Cortex-M0 processor is a 32-bit RISC processor capable of 0.9 DMIPS/MHz that implements a small instruction set architecture (ISA) that consists of less than 60 instructions. This simple ISA is a superset of the 16-bit Thumb ISA first implemented in the ARM7TDMI processor, and has subsequently underpinned the ISA of every ARM processor developed since (for more information on the ISA, see Conclusion).

The reason for the enduring popularity of Thumb lies in its code density - this is crucial in embedded devices where the memory footprint can be the most significant proportion of the silicon cost. The small gate count of the Cortex-M0 (12K gates in the minimum configuration) makes it ideal for low cost devices on the larger silicon process technology nodes used in the manufacture of microcontroller and mixed-signal devices (for example 0.35 μ m, 0.25 μ m, 0.18 μ m). Incredibly the ARM Cortex-M0 can achieve comparable performance efficiency to the ARM7TDMI in just a third of the size and power.

Designing for Ultra Low-Power

Minimizing device power consumption. Within a microcontroller device the processor logic accounts for a proportion of the total current drawn in active or sleep states. In a typical configuration, running at 1.8 Volts on a 180UULL (Ultra Low Leakage) process implemented using ARM standard cell libraries the ARM Cortex-M0 processor contributes as little as 50 μ A/MHz to the active current (depending on the implementation approach and process technology used).

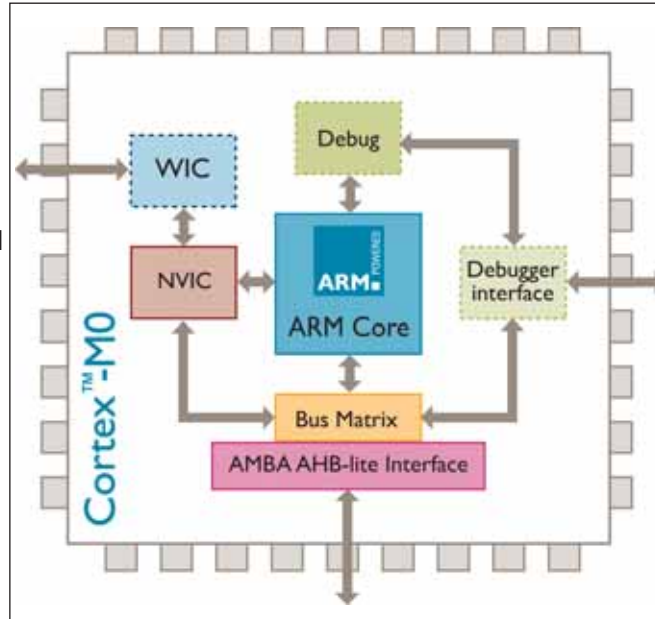
The processor is not the only consumer of power within a device, and there are a many other functions within the microcontroller that must also be carefully tuned to achieve truly ultra low-power (memory and peripheral system

organization and implementation, clock generation, voltage scaling, etc.) - this is the domain of ARM silicon partners and will be covered in the next section. However the processor also has a significant effect on the power consumed by peripherals and memory which will be accessed during activity periods (for example, the smaller code size possible with Thumb can reduce power consumed by flash memory access).

Minimizing the active current also broadens the type of energy source applicable for the applications, enabling energy to be

sourced from smaller, cheaper batteries and also potentially renewable energy harvesting sources. Reducing power consumption can also have the benefit of reducing the power supply complexity.

Reducing System Power. Digital processing at the sensor node can be an effective strategy in reducing the energy consumption of a system. For example, compression, filtering or analysis of analog sensor sample data at the node can significantly reduce the activity of the RF transceiver in a IEEE 802.15 wireless sensor.

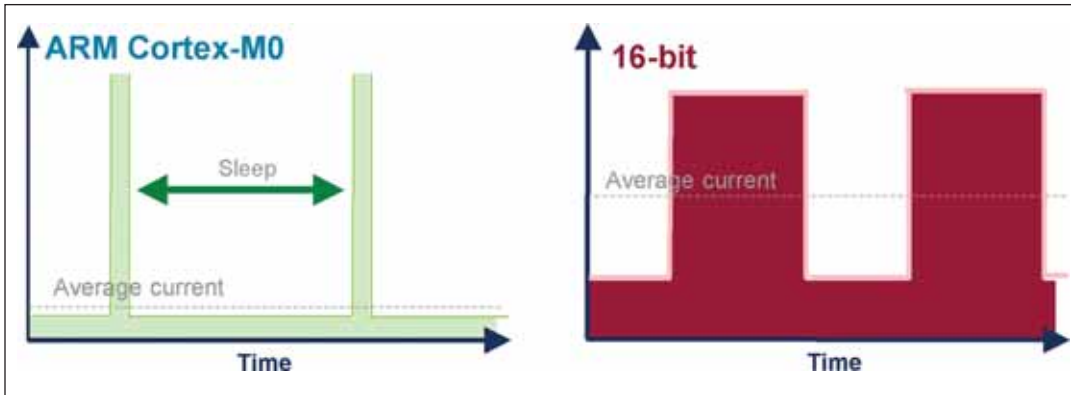


In an optical heart rate monitor application this can mean that only a BPM (beats per minute) value needs to be communicated wirelessly, not the entire sensor sample stream. Another example is interpreting data from an analog sensor to ensure the higher performance applications processor within a system is only woken when the user needs to interact with it (e.g., waiting for a deliberate touch on the screen before communicating input to the host processor and waking a smart phone). In automotive applications, the increasing number of sensors and actuators present in a vehicle means the CAN bus is rapidly reaching capacity - increasing device intelligence can alleviate this through a reduction in communication traffic. These are all applications that can greatly benefit from increased processing capability that the Cortex-M family of processors can provide when closely coupled with an analog sensor source.

Measuring Energy Efficiency. A common strategy to reduce the energy consumption of a microcontroller is to put it into a lower power sleep mode whenever possible, and wake it only when necessary. The energy consumption is considered as the average current of these different activity and sleep states.

(Figure 1, page14, illustrate the concept; in practice the active duty cycle can be lower than 0.05% in an IEEE 802.15 based wireless system, and the active current may be exponentially larger than the sleep current of a device.)

the operation in less bytes of code. Although most users will chose to use C, it is interesting to note how simple and elegant the Cortex-M0 assembler looks in comparison to the 8-bit and 16-bit solutions. With higher precision 24-bit computation the



ARM Cortex-M0 code would remain the same - but the 8-bit and 16-bit assembler becomes increasingly complex, requiring more clock cycles.

ARM Cortex-M0 holds at least a 2x advantage for over 16-bit architectures – meaning the Cortex-M0 device can get back to an ultra low-power sleep twice as quickly.

To estimate the average current there are three variables that must be considered; the *Active Duty Cycle* (i.e., what percentage of the time is the device active), the *Active Current* and the *Sleep Current*. When choosing a low-power microcontroller the processor has an impact on these variables. In the following sections we will see how ARM Cortex-M0 has been designed to address all three.

Working Smarter, Sleeping Longer A significant benefit of ARM Cortex-M0 (and the higher performance Cortex-M3) over 8- or 16-bit architectures is performance efficiency – the ability to complete tasks faster and therefore reduce activity periods. The performance advantage stems from Cortex-M0 being able to perform single cycle 32-bit arithmetic and logic operations (including single cycle 32-bit multiplication) and also perform 8-bit, 16-bit or 32-bit data transfers with indexed addressing in a single instruction. This can have a dramatic effect in reducing the processor clock frequency required, and furthermore also reduces the memory required to store the associated program.

The efficiency advantage of Cortex-M0 over 8 and 16-bit processors also has benefits on the analog sensor capability of the device – the reduced clock frequencies possible mean lower noise, enabling higher precision analog. The reduction of electromagnetic interference accompanying this is also important when considering RF applications. Finally, for the software developer there are many benefits from using a well established and supported architecture, which we will discuss in the later sections of this paper.

Smaller code size and greater efficiency - 16-bit multiply example. Many ADCs sample at 10-bit or 12-bit precision, for which it necessary to transfer and manipulate 16-bit values. In Table 1, we consider a 16-bit x 16-bit multiply and show that ARM Cortex-M0 is not only more efficient, but also encodes

8-bit (8051)	16-bit	ARM Cortex-M0
MOV A, XL ; 2 bytes MOV B, YL ; 3 bytes MUL AB; 1 byte MOV R0, A; 1 byte MOV R1, B; 3 bytes MOV A, XL ; 2 bytes MOV B, YH ; 3 bytes MUL AB; 1 byte ADD A, R1; 1 byte MOV R1, A; 1 byte MOV A, B ; 2 bytes ADDC A, #0 ; 2 bytes MOV R2, A; 1 byte MOV A, XH ; 2 bytes MOV B, YL ; 3 bytes MUL AB; 1 byte ADD A, R1; 1 byte MOV R1, A; 1 byte MOV A, B ; 2 bytes ADDC A, R2 ; 1 bytes MOV R2, A; 1 byte MOV A, XH ; 2 bytes MOV B, YH ; 3 bytes MUL AB; 1 byte ADD A, R2; 1 byte MOV R2, A; 1 byte MOV A, B ; 2 bytes ADDC A, #0 ; 2 bytes MOV R3, A; 1 byte	MOV R4,&0130h MOV R5,&0138h MOV SumLo,R6 MOV SumHi,R7 (Operands are moved to and from a memory mapped hardware multiply unit)	MULS r0,r1,r0
Time: 48 clock cycles Code size: 48 bytes	Time: 8 clock cycles Code size: 8 bytes	Time: 1 clock cycle Code size: 2 bytes

Table 1

Ultra Low-Power

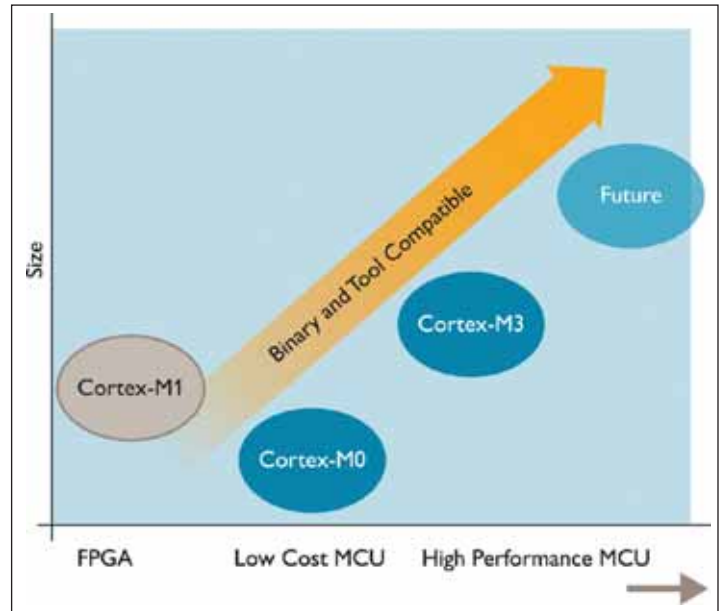
Low active current. One thing that sets ARM Cortex-M0 apart from other 32-bit processors is that in addition to performance efficiency it also offers significantly lower active power (85 μ W/MHz on 180UULL). This is achieved by a combination of an extremely optimized instruction set and micro architecture, and further enhanced by implementation using ARM Physical IP. Minimizing the active current means the device can be driven by smaller and cheaper batteries, and also viably powered by energy harvesting technologies. Although the processor is not the only logic drawing power in a device such as a microcontroller, the increasing cost of energy and demand for low-power devices means any every μ A is significant.

Ultra low-power sleep. The tiny silicon area of the ARM Cortex-M0 processor (just 0.25mm² on 180UULL) means the sleep current drawn by the processor within a microcontroller device is also minimal. As some applications will be asleep over 99% of the time, this is also important contributor to energy efficiency. Like the Cortex-M3, Cortex-M0 has architected support for sleep modes, and software has the ability to put processor to a low-power state and wait for an interrupt to re-awaken it via the Wait For Interrupt (WFI) instruction. These sleep states can take advantage of the state retention technology available in the PMK (Power Management Kit), meaning that internal processor state is preserved and wake-up time can be almost instantaneous. Reducing the wake-up time overhead of the processor and its response to external events via interrupts is critical to minimizing activity. The inclusion of the tightly integrated Nested Vectored Interrupt Controller (NVIC) enables the Cortex-M0 to achieve a low latency, deterministic interrupt response. The sleep on exit feature of the processor means that if Cortex-M0 is woken to service an interrupt it will automatically return to sleep once the ISR has completed.

Simplicity and compatibility

The Cortex-M0 incorporates technologies first introduced in the ARM Cortex-M3 processor to enable faster software development and ensure a binary upwards compatible roadmap to devices based on ARM Cortex-M3 and beyond. These features are common to every processor in the Cortex-M processor family.

Software developers targeting Cortex-M0 do not need to have a deep knowledge of the processor or even write any assembler code at all. For example, in traditional architectures Interrupt Service Routines (ISRs) require an assembler First-Level Interrupt Handler (FLIH) to handle prioritization, context switching and to call any C function second-level handler. In contrast, the Cortex-M0 NVIC performs this first-level handling in hardware, enabling lower interrupt latency and zero jitter response. The huge benefit to the software developer is that they can develop Cortex-M0 interrupt services routines directly in C.



This reduces code size and complexity, and furthermore removes the requirement for assembler programming. The simple and linear Cortex-M0 address space contains no data pages or code pages, meaning memory can be accessed simply and directly for any location in any conceivable device flash or SRAM size. A standard hardware timer (SysTick) is also integrated into Cortex-M0 which makes Real-time Operating System (RTOS) porting between Cortex-M0 processor-based MCU devices much easier by removing the need to make changes to the system timer code of the RTOS.

The ARM Cortex-M0 ISA is based on the 16-bit Thumb instruction set. A few Thumb-2 system instructions are also included in the Cortex-M0 ISA for power management (e.g., Wait For Interrupt) and upwards compatibility. To the developer Thumb means a smaller software code footprint software, reducing the amount of memory required on the device, another significant advantage over 8 and 16-bit architectures.

Looking to the future, the ARM Cortex-M family of processors provides a binary and tools upwards compatibility path for those using the Cortex-M0 who may need more performance longer term, with Cortex-M3 providing an ideal next-step for higher performance and functionality. For those wanting to prototype in FPGA, the ARM Cortex-M1 implements an identical ISA to the Cortex-M0 processor and is 100% binary compatible. There are differences between the Cortex-M1 and Cortex-M0 processors in terms of timing as the two processors have been highly optimized for two different implementation technologies (FPGA and ASIC), but the two processors are instruction set compatible. The Cortex-M family roadmap, combined with the broad software development tools and OS support of the ARM Connected Community provides a significant advantage for those wanting to reuse their code and expertise across MCU, SoC and FPGA developments.

Thumb

User assembly code, compiler generated

ADC	ADD	ADR	AND	ASR	B
BIC	BL		BX	CMN	CMP
EOR	LDM	LDR	LDRB	LDRH	LDRSB
LDRSH	LSL	LSR	MOV	MUL	MVN
ORR	POP	PUSH	ROR	RSB	SBC
STM	STR	STRB	STRH	SUB	SVC
TST	BKPT	BLX	CPS	REV	REV16
REVSH	SXTB	SXTH	UXTB	UXTH	

Thumb-2
System, OS

NOP	
SEV	WFE
WFI	YIELD
DMB	
DSB	
ISB	
MRS	
MSR	

Conclusion

We have seen how the ARM Cortex-M0 helps minimize activity periods and active/stand-by currents, delivering substantial benefits in terms of low peak current and battery life. These technical benefits combined with the software development tools and compatibility roadmap offered by ARM and its microcontroller partners make ARM Cortex-M0 ideal for the next generation of MCU and precision mixed signal devices in long-life battery powered and energy harvesting powered applications.

END

Only Embedded Developer
lets you compare more

than  and .... You

can compare s

And devices and tools.

Then you can buy them. (Wow).

www.embeddeddeveloper.com

One Stop. Shop.

And now, try the new Embedded Developer.cn, the new Chinese site.

EMBEDDED DEVELOPER .COM
FIND. COMPARE. BUY.

The NXP LPC1100 ARM Cortex-M0

By Rob Cosaro, NXP Semiconductors



NXP introduced the first flash-based microcontroller family in early 2004 based on the ARM7 core processor. Since that time NXP's ARM family of microcontrollers has grown to cover a wide range of application spaces and includes the ARM968, ARM926 and the Cortex-M3 core processor families. All of these families have a rich set of peripherals such as Ethernet, USB, CAN, and motor control.

The newly announced Cortex-M3 family called the LPC1700 can operate up to 100 MHz out of zero-wait-state flash. In addition this family comes in many different memory configuration as well as peripheral options to meet the end users cost and performance constraints. Even though these microcontroller families are cost competitive and clearly out-class the high-end 8-bit and 16 bit controllers, designers have the notion that even the Cortex-M3 family is overly complicated for many of their applications. As an example, the programmer models for many 8 and 16 bit controllers are fairly simple with only one mode, not three as in the case of ARM7 or two for Cortex-M3 processors. In addition, the instruction sets are relatively simple with most having less than 80 instructions. A Cortex-M0 based microcontroller is a device that gets back to the roots of the ARM instruction set. The M0 has only one mode and executes a subset of Thumb instructions. The instruction set architecture is composed of less than 60 Thumb

instructions. By using the core of the ARM instruction set a M0-based microcontroller is also upward compatible with Cortex-M3 and is binary compatible with ARM7 processors. An M0-based microcontroller family provides a clear path for users wanting a simpler core while preserving the future software compatibility to the more powerful M3-, ARM7-, and ARM9 processors-based products.

Performance

An M0 microcontroller easily exceeds the performance of high-end 8- and 16-bit devices. The M0 core is rated at 0.9 DMIPS/MHz which is 2 to 4 times the performance of its nearest 8- and 16-bit competitor. DMIPS and MIPS are not always the best indicator of how a user's application will perform, but figure 1 below shows relative performance based on some common routines. Most of the commonly used M0

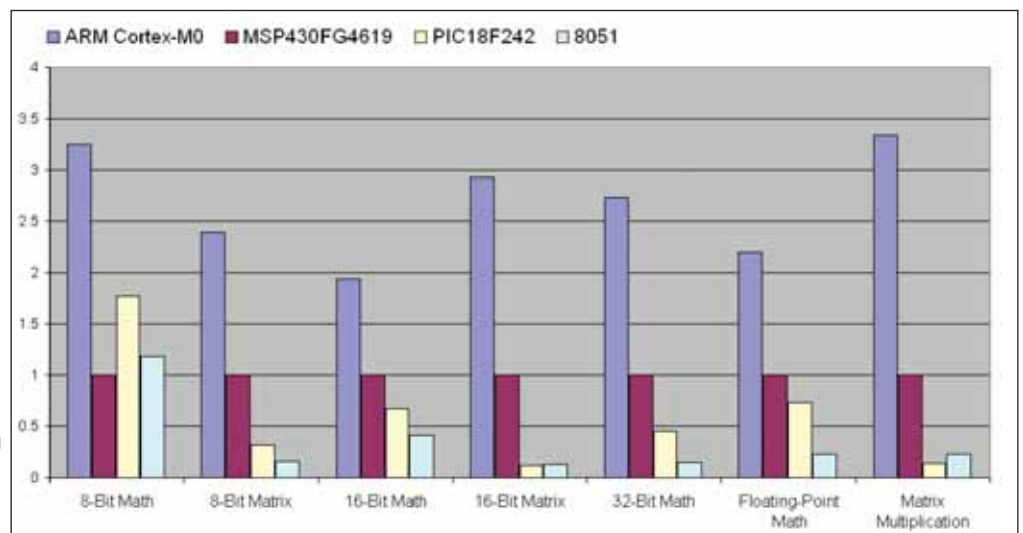


Figure 1: Relative efficiency of ARM Cortex-M0 versus 16-bit and 8-bit architecture

instructions are single cycle and all 8-, 16-, and 32-bit data transfers are achieved in one instruction. Multiply is always clock intensive when dealing long words on an 8- or 16-bit machine. However, since the M0 is based on a 32-bit architecture, NXP's implementation of the M0 can multiply two 32-bit words in a single cycle using a 32 x 32 bit hardware multiplier that is linked directly to the core using the MULS instruction. Performance of a microcontroller is not just about execution speed but also how well it deals with interrupts. Interrupt performance can be characterized by latency and jitter. Latency is defined as how long it takes for the processor to enter an interrupt service routine and jitter is defined as the variability of the latency. The M0 minimizes latency by tightly coupling the interrupt controller to the core.

The result is a fixed latency of 16 cycles for the highest priority interrupts. To reduce latency on nested interrupts there is a built-in mechanism that avoids having to restack when a higher priority interrupt arrives before the previous interrupt has entered the service routine. In addition, there is support for tail chaining which reduces interrupt latency by folding together an exception exit unstacking sequence with the subsequent exception entry stacking sequence, allowing direct entry into the ISR. The interrupt controller supports up to 32 different interrupt sources and includes a non-maskable interrupt input. As with all 8- and 16-bit microcontrollers, the interrupt priority and the saving of context is all handled by hardware. This eliminates the need for writing any assembly code for dealing with interrupts.

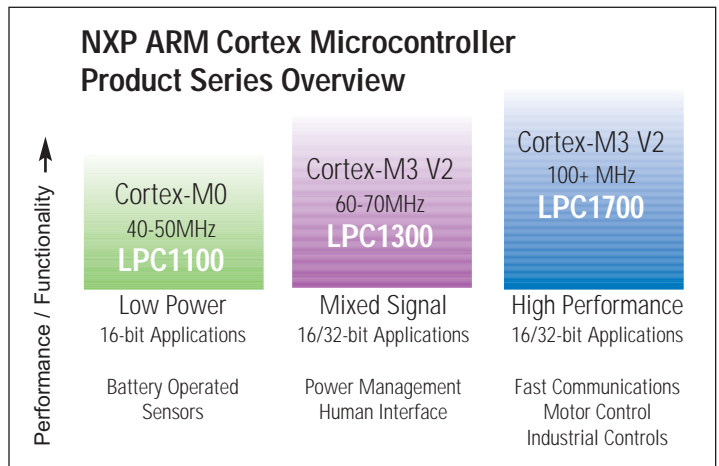
Power A common misconception is that only 8- or 16-bit machines can be low-power, because of their reduced logic content. With M0 this is not the case. In fact, the gate count for the M0 is lower than many 8051 implementations. The M0 core achieves power consumption numbers of less than 40uA/MHz. In addition, current consumption is not a good measure of how much work that can be done by the core with given energy. Since the M0 core is based on a 32-bit architecture it can use the current more efficiently than either a 8- or 16-bit architecture.

“ The licensing agreement enables NXP to continue to lead in the MCU market place and provide a ARM technology-based processors to meet various performance, power and price needs. The Cortex-M0 processor is the new entry-level solution offering customers ease of use through a very small Thumb compatible instruction set, and low cost industry standard development tools. ”

Geoff Lees, VP and General Manager,
Microcontroller Division, NXP Semiconductors

A Perfect fit for the ARM-based LPC Family

For the same calculation performance the core can run at 2 to 4 times slower, which will save half or one quarter of the power for the same application. Or in plain English in other words, the M0 microcontroller can finish the work faster and spend more time in one of the power-down modes. In the end, this translates to longer battery life for the application.



An M0-based LPC family of microcontrollers provides a perfect match to the existing family. It provides an extremely low-power platform based on the existing 32-bit ARM instruction set architecture. The M0-based platform will concentrate on the low-power space dominated by other existing architectures.

Tool Chain Support Since the M0 uses a common bus architecture it is simple to keep the peripheral set constant across the LPC families. This makes it easier for code to be ported across families. The LPC family that is based on the M0 core uses a simple peripheral set that makes it ideal for most applications that use lower performance cores.

Since the M0 processor uses the Thumb instruction set and a subset of the ARM coresight debug infrastructure, existing tool chains can easily support the M0-based LPC microcontroller family. The M0-based LPC family will use the smaller-footprint serial wire debug (SWD) interface to allow for more user pins. In addition, it will be possible to use the debug pins as GPIO as well. The M0-based LPC devices will include 4 break points as well as 2 watch points to allow for more effective debugging.

Conclusion There are many users of 8- and 16-bit microcontrollers that are reluctant to use 32-bit architectures citing either overkill or complexity. The M0 is an architecture that makes this argument irrelevant. The M0-based LPC family provides a microcontroller that is very low-power, has better real-time performance than microcontrollers of lower bit width and provides a bridge to the full spectrum of the LPC families.

END

Low Power Design using the LPC1100 Series

By Rob Cosaro, NXP Semiconductors

This paper discusses the low-power characteristics of the M0-based LPC1100 microcontroller series as well as system design techniques that minimize the amount of energy consumed from a power source.

Microcontroller low-power design considerations

Definitions:

In order to understand the power consumption of a microcontroller, it is important to understand the basic components of dissipation of a CMOS device. There are two major categories of consumption; dynamic consumption and static consumption.

Dynamic power consumption

Dynamic power consumption of a CMOS device to a first order is defined as

$$P = \frac{1}{2} \cdot f \cdot C \cdot V^2$$

Where f = operating frequency

C = Load Capacitance

V = Supply Voltage V_{DD}

Figure 1 shows a diagram of a simple CMOS inverter. When this inverter switches, it must charge or discharge the load capacitance, which dissipates power.

The load capacitance is a combination of the interconnect capacitance and the gate capacitance of all the devices it is driving. If the device is not switching, all that is consuming power is the leakage current of the device.

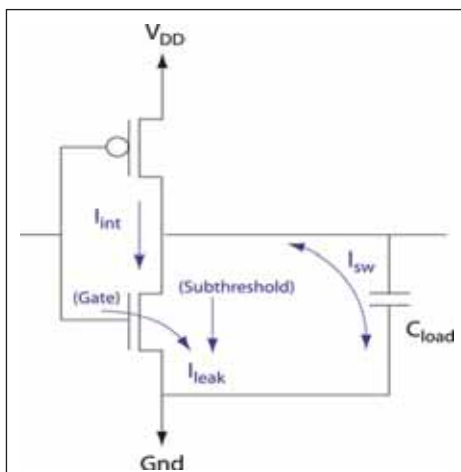


Figure 1: CMOS dissipation

Therefore, for a given process geometry the dissipation varies as the square of the voltage and linearly with frequency. The characteristic that power consumption varies linearly with frequency gives rise to a commonly quoted number for microcontrollers, which is current consumption per MHz. For low-power devices this is given as $\mu\text{A}/\text{MHz}$, and ranges from $200\mu\text{A}/\text{MHz}$ to over $300\mu\text{A}/\text{MHz}$. These numbers are somewhat misleading since there is no standard on how the measurements are taken. The key point is how much work is performed for the current consumed or, for a more comprehensive measurement, is how much energy is consumed for a given calculation. Since this type of measurements is not widely used yet the $\mu\text{A}/\text{MHz}$ metric is used in this discussion.

The amount of current used per MHz by the digital CMOS structures is not the only aspect of the current consumed by the device there are analog circuits that are required to support the digital domain.

These can be classified into the timing components, power control components, memories and peripherals. The timing, power control and memory components are part of the microcontroller platform and are not optional, but the analog peripherals are part of the feature set and will differ across the microcontroller family.

Table 1 shows the timing components used in the LPC1100. The table is arranged from lower to higher power consumption. As with all analog design there is a trade-off between accuracy and the amount of current it consumes. The LPC1100 has a flexible scheme for controlling these components that can trade off consumption versus accuracy so they can be tailored to the application.

Timing components	Attributes
Low power internal osc	Low frequency, Low accuracy, Low power
Internal RC oscillator	Accurate better than 1%
Crystal oscillator	High accuracy low jitter.
DLL	Fast start-up, higher jitter, low consumption
PLL	Slow startup, low jitter. Moderate consumption

Table 1: LPC1100 timing components

Table 2 describes LPC1100 components that control the power to the LPC1100. As with the timing components these can also be tailored to the applications requirements.

Power control components	Attributes
Power management Unit (PMU)	Provides power control for power modes.
Power on reset POR	Low power. Less accurate
Brown out detect (BOD)	Accurate, multi-level

Table 2: Power control components

The current consumption for the core is not just about the slope but also the offset current that comes from the analog components that are required to support the core. This sometimes is referred to as the zero-hertz current. Since the LPC1100 has a flexible clocking architecture this current is not fixed. As the frequency is lowered, switching off the clocking components that are not required to generate the required operating frequency can reduce the offset current. As an example, the LPC1100 can be operated on the loose low-power oscillator from 0 to 1 MHz and then the more accurate internal RC oscillator can be turned on to provide the frequency from 1 to 12 MHz.

Leakage consumption Leakage consumption is the current that the CMOS junction draws when the digital logic is not switching. This current is highly dependent on the process node and then on how the libraries in the node are optimized. For the LPC1100 the libraries are optimized for low leakage. Giving the user different power down options can further optimize leakage. Besides the leakage of the CMOS junctions, various analog features can also be controlled in these modes.

Power mode	Description
Sleep mode	Core clocks off, Peripherals can be left on
Power-down mode	Clocks off, Optional analog features off
Deep power down mode.	Power turn off. Always on domain registers are active.

Table 3: Power modes for the LPC1100

Sleep Mode Sleep mode turns the clocks off to the core but the user has the option to leave on peripherals. The power in this mode is not just leakage but dynamic current of the peripherals that are left on. In this mode data can be still received, but the core retains its state and can continue operation when required.

Power Down-mode All clocks to the digital logic are turned off and the analog sub-systems can be controlled to have flexible wake up times depending on the application requirements. The lowest power mode is when all the analog clocking elements are turned off. Wake up time is determined by the selection of the wake up clock source. The fastest time is from the low-power oscillator and the slowest time is from the crystal oscillator and the PLL.

Deep Power Down-mode In this mode power is turned off to the internals of the microcontroller, except a small always-on domain. The always-on domain has a set of registers that can store the information about what happened before the microcontroller when into deep power-down mode. Wake up from this mode occurs either through a wake up pin or reset.

Code Efficiency The LPC1100 uses the new Cortex-M0 core from ARM, which has a big impact both on dynamic current as well as leakage current. Focusing on a simple instruction set reduces dynamic current. The M0 mostly uses Thumb instructions. These are 16 bits wide and are interpreted by the core as a 32-bit instructions. The core also uses a simplified bus interface to reduce gate count and minimize clocking. In addition, the core is architected to take advantage of clock gating and simplified library elements. Taking all this into account the core is rated at less than 70uA/MHz. As was previously stated, this number is somewhat meaningless since it doesn't include any information about how much work can be done with this current. However, the M0 core is rated at 0.9 DMIPS/MHz, which is a higher rating than an ARM7 core. Using this core further enhances the leakage current since the gate count is equivalent to 8- and 16-bit cores. Since leakage current is proportional to gate count any savings in core logic has a big impact.

Low-power System Considerations

How the microcontroller power modes are used depends on the application. If the power source is always there but has limited capacity, the microcontroller may be always clocked. The LPC1100 can change frequency on the fly depending on processing demand. The LPC1100 current consumption at 30 MHz is specified at 6mA. This can be reduced to a little over 200uA when running at 1 MHz on the low-power internal oscillator.

However, many applications that need to minimize consumption must rely on the power-down and deep power-down modes. These applications spend most of their time in a quiescent state waiting to process data. The processor must wake up quickly, process the required data, and then go back to the quiescent state. Many of these applications are battery-powered where a low average current is important to extend battery life. In order to lower the average current it is important to be able to process the data as quickly as possible to reduce the duty cycle. Since the M0 is a 32-bit processor it can perform the calculations much quicker than small width processors.

Figure 2 on the next page, shows how processing performance effects the average current. The figure assumes the peak current and the power-down currents are the same for the various processor types. The M0 core has the capability to be one half to one fourth the average current of processors of lower bit widths. The M0 allows the LPC1100 to achieve peak currents of 200uA/MHz.

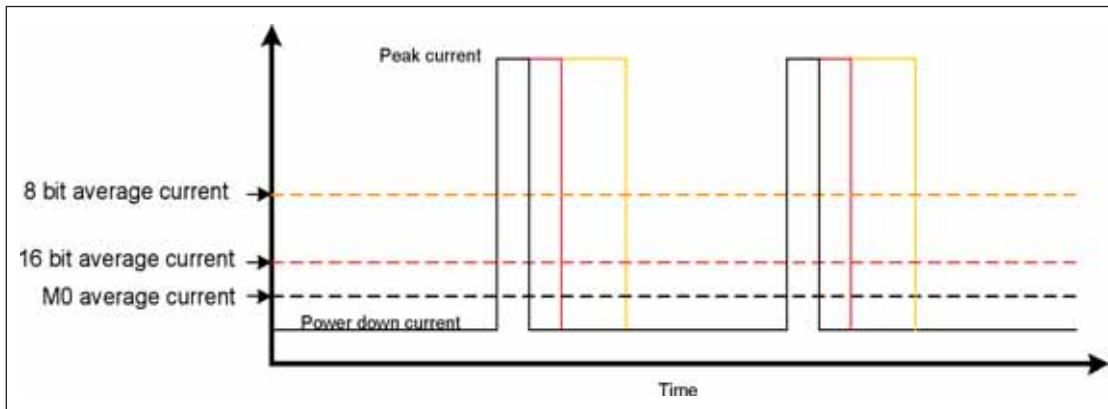


Figure 3: Average Current

Low average current is critical to extend battery life. This means low quiescent current and small duty cycles. The LPC1100 has deep power-down current of less than 300 nA and peak currents of 200uA/MHz. Figure 3 shows the effects duty cycle has on battery life. The battery used for these calculations is a 230mAh lithium button cell. This plot shows the effect that quiescent current has on battery life and the kind of duty cycles that are required to exceed three years. The average current assumes a peak current of 2mA which means the LPC1100 is operating at 10 MHz. It also includes

the effects of startup time since lowering quiescent current extends start up time. If the LPC1100 deep power-down mode is used then for 1 ms of processing time out of a period of 200ms a 3 year battery life is possible.

Conclusion The M0 core allows the LPC1100 to achieve power numbers that are consistent with current low-power microcontrollers,

but the processing power of the core is much higher than the current 8- and 16-bit processors. This extends battery life by lowering average current for battery powered applications and reduces the power requirements in other applications. The M0 based LPC1100 will have a significant impact on the capabilities of low-power applications.

END

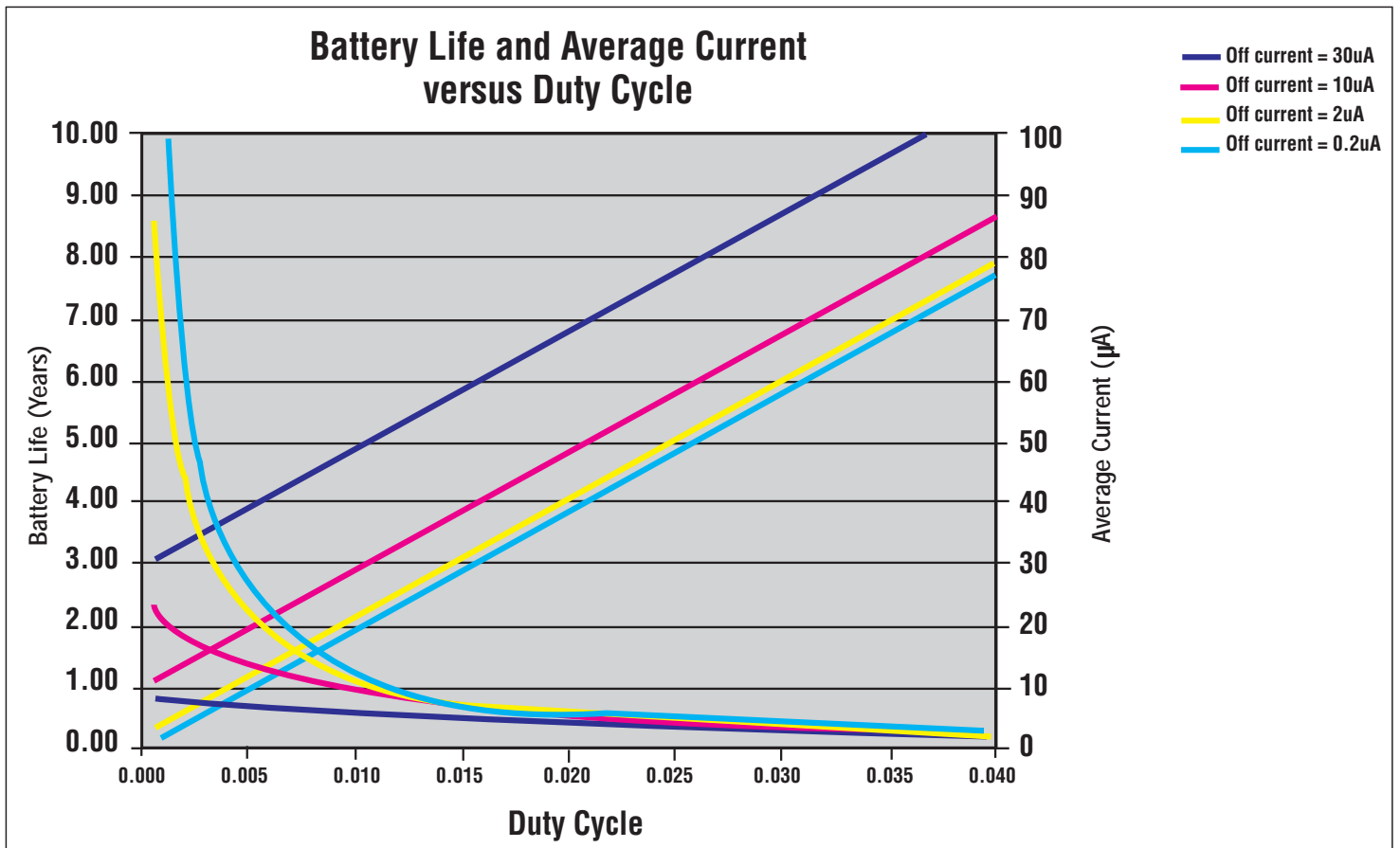


Figure 4: Effect duty cycle has on battery life



NXP 50-MHz, 32-bit Cortex-M0™ microcontrollers LPC1100

Cortex-M0 based microcontrollers with industry-leading power and efficiency

Built around the new Cortex-M0 architecture, the smallest, lowest power, and most energy-efficient ARM core ever developed, these MCUs are ideally equipped for use in battery-powered consumer devices, smart meters, motor control, and more.

Key features

- ▶ ARM Cortex-M0 processor
 - 50-MHz operation
 - Nested Vectored Interrupt Controller for fast deterministic interrupts
 - Wakeup Interrupt Controller allows automatic wake from an priority interrupt
 - Three reduced-power modes: Sleep, Deep-sleep, and Deep power-down
- ▶ Memories
 - Up to 128 KB Flash memory
 - Up to 16 KB SRAM
- ▶ Serial Peripherals
 - UART with fractional baud rate generation, internal FIFO, and RS-485 support
 - SPI controller with FIFO and multi-protocol capabilities
 - I²C-bus interface supporting full I²C-bus specification and Fast mode plus with a data rate of 1 Mbit/s with multiple address recognition and monitor mode

Analog Peripherals:

- 10-bit Analog-to-Digital Converter with eight channels and conversion rates up to 250K samples per second

Other peripherals:

- Up to 42 General Purpose I/O (GPIO) pins with configurable pull-up/down resistors and a new, configurable open-drain operating mode
- Four general purpose counter/timers, with a total of four capture inputs and 13 match outputs
- Programmable Watchdog Timer (WDT) with lock-out feature
- System tick timer
- Each peripheral has its own clock divider for power savings

Applications

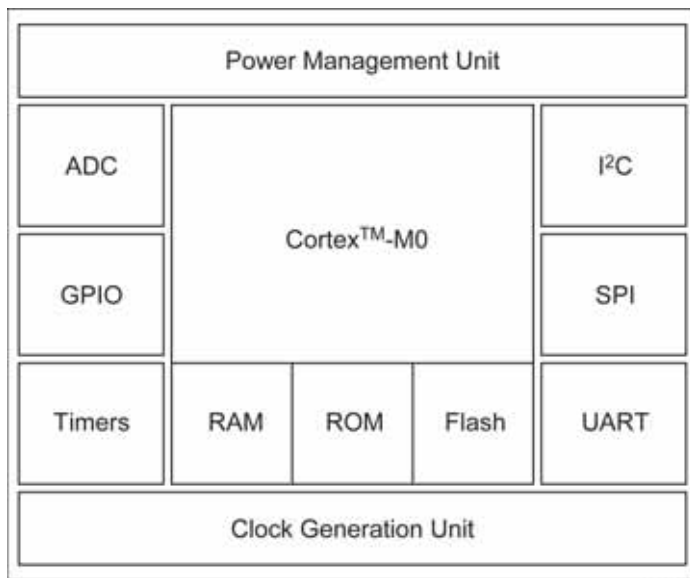
- ▶ Battery-powered systems
- ▶ e-Metering
- ▶ Consumer peripherals
- ▶ Remote sensors
- ▶ 16-bit applications

The LPC1100 is the world's first Cortex-M0 based microcontroller series offering users a cost effective, very easy to use 32-bit MCU which is code and tool compatible with other NXP ARM based MCU products. With 32-bit performance combined with multiple power modes and very low Deep sleep power, the LPC11xx offers industry leading energy efficiency greatly extending battery life. The LPC11xx sets new benchmarks in performance efficiency with dramatically improved code density enabling longer battery life and lower system costs.

Third-party development tools

Through third-party suppliers, we offer a range of development and evaluation tools for our microcontrollers. For the most current listing, please visit www.nxp.com/microcontrollers.

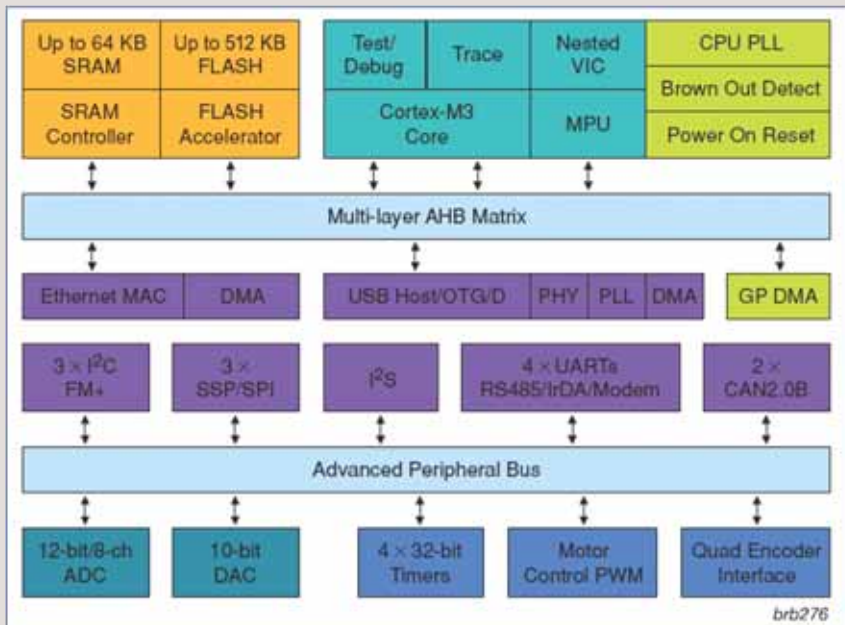
LPC1100 block diagram



00000000

Additional features

- ▶ Serial Wire Debug and Serial Wire Trace Port
- ▶ High-current output driver (20 mA) on one pin
- ▶ High-current sink drivers (20 mA) on two pins
- ▶ Integrated PMU (Power Management Unit) to minimize power consumption during Sleep, Deep-sleep, and Deep powerdown modes
- ▶ Single 3.3 V power supply (1.8 V to 3.6 V)
- ▶ 15 GPIO pins can be used as edge and level sensitive interrupt sources
- ▶ Cock generation unit with divider that can reflect the main oscillator clock, IRC clock, CPU clock, and Watchdog clock.
- ▶ Processor wake-up from Deep-sleep mode via interrupts from various peripherals
- ▶ Brownout detect with four separate thresholds for interrupt and forced reset
- ▶ Power-On Reset (POR)
- ▶ Crystal oscillator with an operating range of 1 MHz to 25 MHz
- ▶ 12 MHz internal RC oscillator trimmed to 1 % accuracy that can optionally be used as a system clock
- ▶ PLL allows CPU operation up to the maximum CPU rate without the need for a high-frequency crystal. May be run from the main oscillator, the internal RC oscillator, or the Watchdog oscillator.
- ▶ Available as 48-pin LQFP package and 33-pin HVQFN package



100 MHz, 32-bit Microcontrollers with Cortex-M3™ core, LPC1700 series

The LPC1700 series of low power cost-effective Cortex-M3 microcontrollers feature best in-class peripheral support such as Ethernet, USB 2.0 Host/OTG/Device, and CAN 2.0B. Operating at speeds up to 100 MHz, they have up to 512 KB of FLASH, up to 64 KB of SRAM, 12-bit A/D and 10-bit D/A converters as well as an internal RC oscillator.

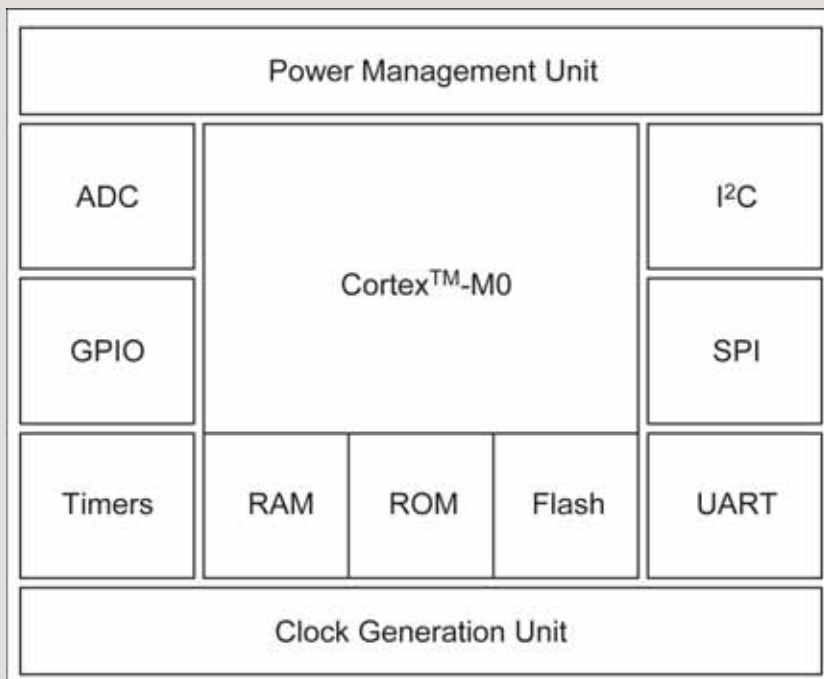
Cortex-M3 based microcontrollers with Ethernet, USB, CAN and 12-bit ADC

NXP develops vibrant media technologies that bring your ideas to life. Our innovative solutions enhance images, sharpen sound and simplify information sharing. Find out more about this family of advanced, low power 32-bit microcontrollers at our website

www.nxp.com/microcontrollers



NXP
founded by Philips



NXP 50-MHz, 32-bit Cortex-M0™ microcontrollers, LPC1100

Built around the new Cortex-M0 architecture, the smallest, lowest power, and most energy-efficient ARM core ever developed, these MCUs are ideally equipped for use in battery-powered consumer devices, smart meters, motor control, and more.

Cortex-M0 based microcontrollers with industry-leading power and efficiency

NXP develops vibrant media technologies that bring your ideas to life. Our innovative solutions enhance images, sharpen sound and simplify information sharing. Find out more about this family of advanced, low power 32-bit microcontrollers at our website

www.nxp.com/microcontrollers

