



## 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 $\mu$ m, 0.25 $\mu$ m, 0.18 $\mu$ 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 $\mu$ A/MHz to the active current (depending on the implementation approach and process technology used).

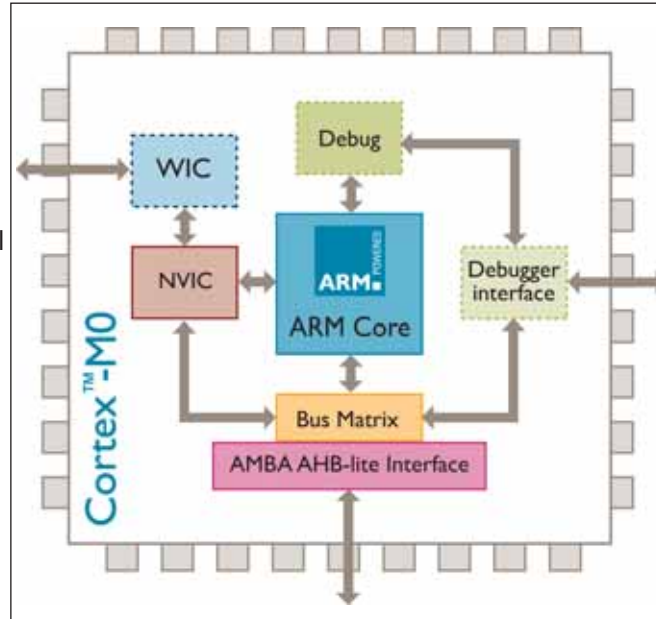
The processor is not the only consumer of power within a device, and there are a many other functions within the microcontroller that must also be carefully tuned to achieve truly ultra low-power (memory and peripheral system

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.

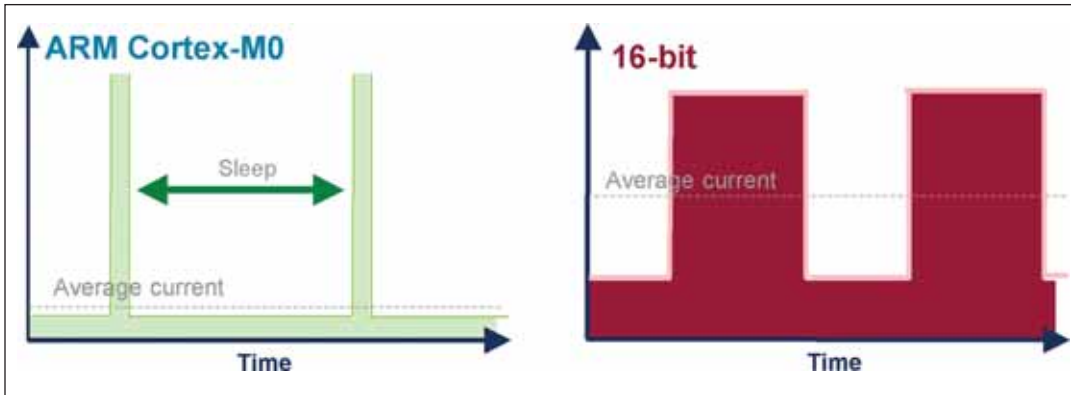


In an optical heart rate monitor application this can mean that only a BPM (beats per minute) value needs to be communicated wirelessly, not the entire sensor sample stream. Another example is interpreting data from an analog sensor to ensure the higher performance applications processor within a system 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

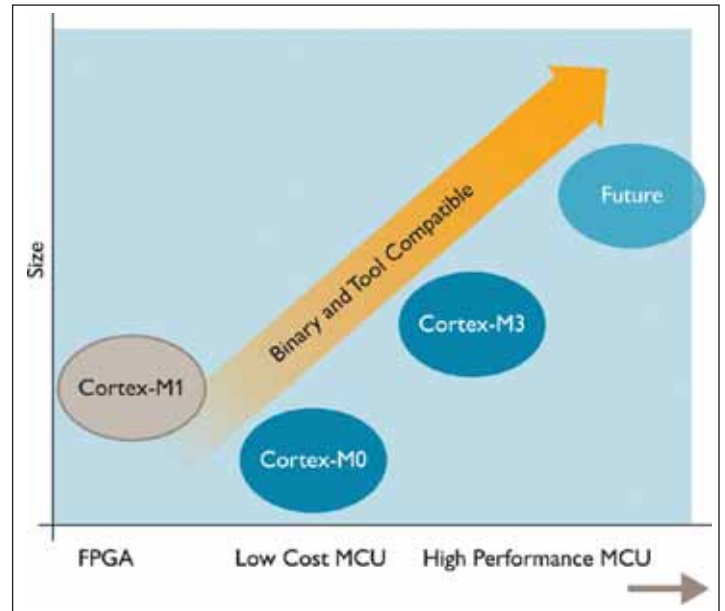
**Low active current.** One thing that sets ARM Cortex-M0 apart from other 32-bit processors is that in addition to performance efficiency it also offers significantly lower active power (85 $\mu$ W/MHz on 180UULL). This is achieved by a combination of an extremely optimized instruction set and micro 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  $\mu$ A is significant.

**Ultra low-power sleep.** The tiny silicon area of the ARM Cortex-M0 processor (just 0.25mm<sup>2</sup> 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](http://www.embeddeddeveloper.com)

One Stop. Shop.

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

**EMBEDDED DEVELOPER** .COM  
FIND. COMPARE. BUY.