

## 32 位低功耗 MCU 设计全面介绍

传统的低功耗 MCU 设计都是以 8 位 MCU 为主,因为 8 位内核闸门相对较少,运行或泄露电流低,售价也相对低廉。但是,许多新兴的应用都需要比 8 位内核更大的处理效率。近年智慧生活的抬头、物联网的建立,手持式消费性电子产品与无线功能需求越来越高、设计越来越复杂,要提高性能的同时又要兼顾低功耗,需要有一高性能低功耗的主控 MCU 来作为平台。另一方面,工业上的智能化也在展开,如远程监控、数字化、网络化等。简单说来,就是人物之连结(云端应用)、物物之连结(物联网)需求越来越多,导致产品功能越来越复杂,运算量越来越高,2009 年 ARM 发表了 32 位 CortexTM-M0 内核,提供 MCU 厂商一个强而有力的平台,加上制程微缩技术的进步,嵌入式闪存制程普及化及降价,主要成本来自内存大小及模拟外围和 IO 管脚数量,CPU 内核的成本差异已大幅缩短,更促进了高性价比 32 位低功耗 MCU 的快速发展。

### MCU 功耗来自何处

在开始讨论低功耗 MCU 设计前,必须先探讨 MCU 功耗的来源,其主要由静态功耗及运行功耗两部分组成。考虑实际的应用,最后决定系统功耗性能指针则必须计算平均功耗。

### 运行功耗

现代 MCU 已整合相当多的的模拟外围,不能单纯考虑数字电路的动态功耗。MCU 运行时的总功耗由模拟外围功耗和数字外围的动态功耗相加而得。模拟电路的功耗通常由工作电压及其性能要求指针来决定,例如 100 ns 传递延迟

(Propogation Delay) 的比较器工作电流可能约为 40 微安,当允许传递延迟规格为 1  $\mu$  s 时,工作电流有机会降到个位数微安。

数字电路的动态功耗主要来自开关频率、电压及等效负载电容,其计算公式如下:

$$P_{\text{Dynamic}} (\text{动态功耗}) \sim f (\text{工作频率}) \times CL (\text{等效负载电容}) \times VDD2 (\text{工作电压})$$

由以上公式可以理解到降低动态功耗最直接的方式为降低工作电压及工作频率。但 MCU 实际应用面通常要求更宽广的工作电压及更高的效能。在降低工作电压方面,可以选择更新近的制程,并透过 LDO 让 CPU 内核、数字电路及与管脚输入输出电压无关的模拟外围在低压操作,IO 管脚及需要与其他外部电路连接的模拟外围则在较高的系统电压操作。如此可以兼顾低功耗及宽工作电压的需求。在降低工作频率这项参数上,一个设计优良的 32 位 MCU 更能突显其效能优势,除了直觉的 MIPS 比较之外,32 位总线也代表更高的数据存取带宽,能以更低的工作频率达到相同的效能,进而降低整体功耗。另外,如果 MCU 内建与操作频率相关的模拟外围,例如石英晶体震荡电路、嵌入式闪存或电流式 DAC,其电流消耗与转换频率成正比,也要纳入低功耗 MCU 的动态功耗设计考虑。

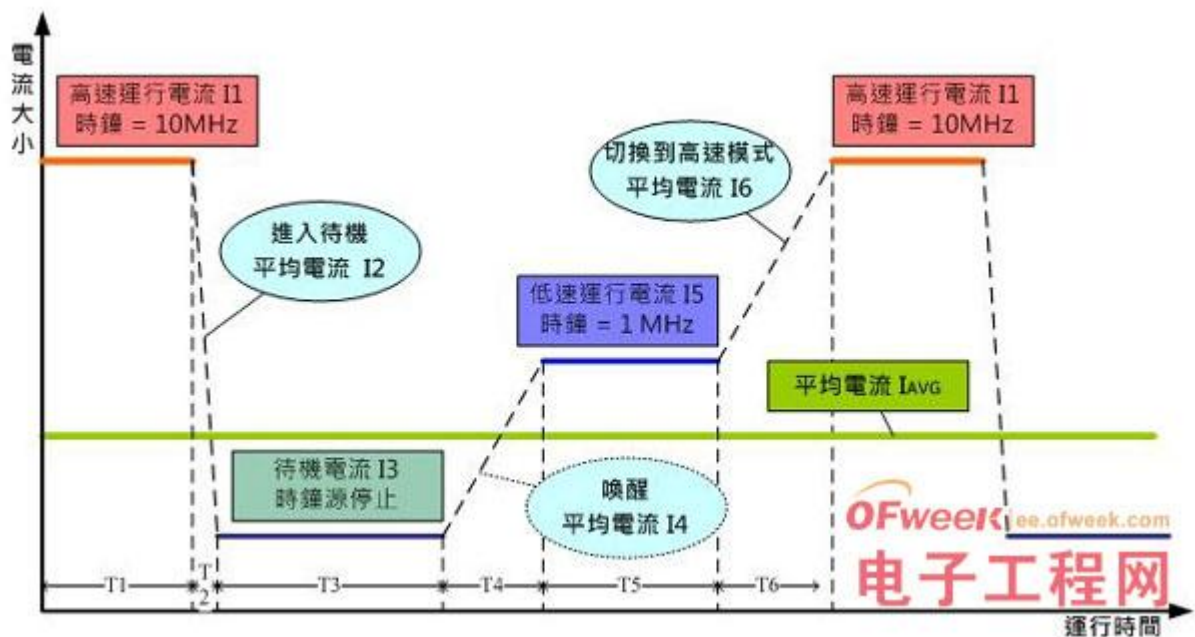
## 静态功耗

传统静态功耗的定义是指系统时钟源关闭时数字电路的漏电流。但是在混合信号低功耗 MCU 的设计中要同时考虑下列多种漏电流来源，包含数字电路漏电流、SRAM 漏电流、待机时已关闭的仿真电路漏电流（例如 ADC, 嵌入式闪存）、待机时不关闭的仿真电路工作电流（例如 LDO、BOD）及 IO 管脚的漏电流。因为时钟源已关闭，影响静态功耗的主要参数为制程、电压及温度。所以降低静态功耗必须选择超低功耗制程，但是低功耗制程通常伴随较高的  $V_t$ ，导致低电压模拟外围设计困难。另外，以 MCU 待机电流 1 微安的规格，代表数字电路漏电 + RAM 保持电流 + LDO 工作电流 + BOD（降压侦测或重置电路）工作电流总和必须小于 1 微安，对于 Flash, RAM 越来越大及功能越来越多的低功耗 MCU 设计厂商而言，是十分艰巨的挑战。

## 平均功耗

在系统级要兼顾低功耗及高效能，必须考虑实际应用面的需求，例如无线环境传感器可能让 MCU 主时钟及 CPU 关闭，只开启低频时钟，定时唤醒外围电路进行侦测，当符合设定条件的事件发生时快速启动 CPU 进行处理，即使没有任何事件发生，也必须定时激活 CPU 维持无线传感器网络的联机。在遥控器的应用中，则可能完全将所有时钟源都关闭，当用户按键时快速唤醒时钟源及 CPU 进行处理。另外，许多应用都会加入一个 MCU 作为主机处理器的协处理器，用于监控键盘或红外线输入、刷新显示器、控制主处理器电源以及智能电池管理等任务。此时平均功耗比单纯的运行功耗或待机功耗更具指标性意义。

平均功耗由下列主要参数组合而成：运行功耗及运行时间，静待功耗及待机时间，不同运行模式之间的切换时间。兹以下图进行说明：



$$\text{平均电流 (IAVG)} = (I1 \times T1 + I2 \times T2 + I3 \times T3 + I4 \times T4 + I5 \times T5 + I6 \times T6) / (T1 + T2 + T3 + T4 + T5 + T6)$$

因为进入待机模式时间很短，忽略此段时间的电流消耗，公式可以简化为：

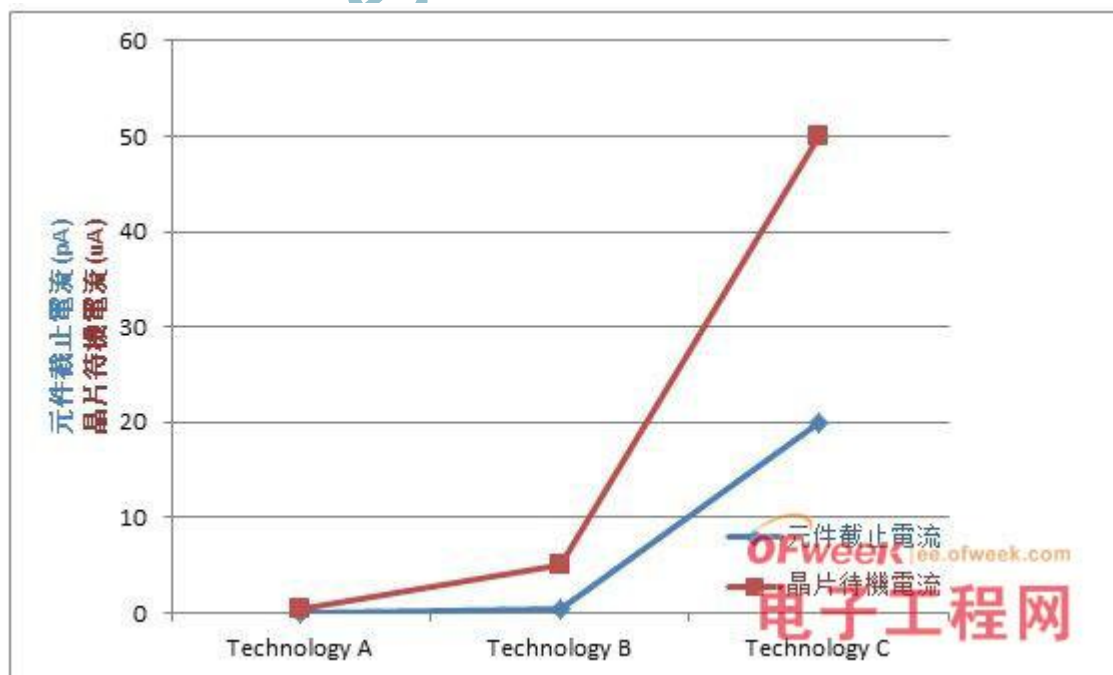
$$\text{均电流 (IAVG)} = (I1 \times T1 + I3 \times T3 + I4 \times T4 + I5 \times T5 + I6 \times T6) / (T1 + T3 + T4 + T5 + T6)$$

由以上公式观察到除了降低运行电流及静态待机电流外，降低运行时间、唤醒时间及高低速运行模式切换时间也是降低整机功耗的重要手段。另外，上图同时指出，低功耗 MCU 支持动态切换运行时钟频率是必要的功能。

## 低功耗 MCU 设计考虑

### 制程选择

为了达到低功耗的运作，并能有效地在低耗电待机模式下，达到极低的待机功耗，可以透过对制程的选择而站上基本的要求门坎。在不强调速度极致的某些制程分类，选择极低组件截止电流制程（如下图）进行逻辑闸制作，并进行数字设计是方法之一。选择这种策略的额外效益是，通常也能在降低动态操作电流上，达到较佳的表现。另外，由于高温大幅增加静态电流，当温度由摄氏 25 度增加到摄氏 85 度时，一个典型比例约增加 10 倍的静态电流，以非低功耗 0.18 微米制程开发的 32 位 MCU，逻辑阀门数 200K、4KB SRAM 在核心电压 1.8V、摄氏 25 度的静态耗电约为 5 ~ 10 微安，当温度升高到摄氏 85 度时，静待电流将会飙高到 50 ~ 100 微安。而低功耗制程在摄氏 85 度仅约 10 微安静态电流。



## 低功耗高效能的 CPU 内核

早期低功耗 MCU 受限于成本及制程技术，大都选择 8 位 CPU 内核，但随着工业上的智能化也在展开，如远程监控，数字化、网络化等。简单说来，就是人物之连结（云端应用）、物物之连结（物联网）需求越来越多，导致产品功能越来越复杂，运算量越来越高，8 位 MCU 已逐渐无法满足效能需求。为了兼顾低功耗高效能，选择适用的 32 位 CPU 内核乃大势所趋。

选择低功耗 CPU 内核，除了单位频率耗电流外，还需要综合考虑紧凑的低内存代码，相同功能所需的代码越长，除了增加内存成本，也代表更长的运行时间及功耗。另外，由于软件开发成本在后期将会越来越高，大量的参考代码及更多的第三方开发商的支持，都可以有效降低软件的开发时间及成本。所以选择一款更多人使用的 CPU 内核也是重要的考虑之一 ARM CortexTM-M0 阀门数仅 27K, 使用的电量在 1.8V, 超低泄漏 180ULL (Ultra Low Leakage) 仅约 50 $\mu$  A/MHz. M0 内核采用 Thumb2 指令集架构，产生出非常紧凑的低内存代码，进一步降低了电源需求。ARM 自 2009 年发表了 32 位 CortexTM-M0 内核以来，包括 NXP、新唐科技、ST、Freescale 等多家国内外 MCU 大厂相继投入 CortexTM-M0 MCU 开发，不论供货或者品种的齐全度都已十分成熟，投入 CortexTM-M0 的 MCU 开发商也在持续增加中。

### 低功耗数字电路

对于一般的同步数字电路设计，要使数字单元有效降低操作电流，透过控制时钟的频率或截止不需要的时钟跳动，也是重要的方法。低功耗 MCU 通常配备丰富的时钟控制单元，可对各别的数字外围单元，依照需求做降频或升频的操作调整，在达到运作能力的同时，用最低的频率来运行。但为了达到更弹性的时钟源配置，可能导致 CPU 内核和外围电路时钟不同步的现象，此时必须仔细考虑电路设计，保证跨时钟领域数据存取的正确性。

另外为了尽量降低 CPU 介入处理时间或降低 CPU 工作频率而节省下来的功耗，可以提供 DMA 或外围电路相互触发电路进行数据的传递，例如 Timer 定时自动触发 ADC 或 DAC, 并透过 DMA 进行数据由 ADC 到 RAM 或者 RAM 到 DAC 的搬移，同时在 ADC 的输入可以增加简单的数字绿波及平滑化电路，如此不须要 CPU 经常介入处理，也不会因为需要实时处理 ADC 或 DAC 事件导致中断程序占用太多时间，降低系统的实时性及稳定性。

### 支持多种工作模式

为了配合不同的应用需求，并达到系统平均功耗的最小化，低功耗 MCU 需要提供多种操作模式，让用户灵活调配应用，常见的操作模式有下列数种：

正常运行模式：CPU 内核及外围正常工作，能实时改变 CPU 及外围的工作频率 (On the Fly) 或关闭不需要的时钟源以获得最佳的工作效能。

低频工作模式：CPU 内核及外围工作于低频的时钟源，例如 32.768K 晶震或内部低频 10K RC 震荡器。通常此时最大的耗电来源为嵌入式闪存及 LDO 本身的耗电流。如果此时的执行程序不大，可以考虑将程序运作于 RAM 以降低平均功耗。请注意并不是所有 MCU 都能支持在 RAM 执行程序。

Idle 模式：CPU 内核停止，时钟源和被致能的外围电路持续工作，直到外围电路符合设定条件唤醒 CPU 进行数据处理或控制执行流程。通常高频的运行模式，CPU 及嵌入式闪存消耗相当大比例的电流，故 Idle 模式能有效降低平均功耗。

待机 RAM 保持模式：CPU 内核及所有时钟源关闭，内建 LDO 切换到低耗电模式，但是 RAM 及 IO 管脚持续供电，维持进入待机之前的状态。

RTC 模式：CPU 内核及高频时钟源关闭，内建 LDO 切换到低耗电模式，由于此时 LDO 供电能力降低，仅能提供低耗电的外围电路运行，例如 32.768K 晶振、RTC（实时时钟计数器）、BOD（降压侦测或重置电路）、TN 单色 LCD 直接驱动电路等。

深层待机模式：CPU 内核及所有时钟源关闭，关闭 RAM 及 LDO、BOD 等所有外围电路的电源，仅 IO 管脚（或部分 IO 管脚）持续供电，由 IO 管脚或重置（Reset）管脚唤醒 CPU。因为此模式下，RAM 的数据已丢失，通常会进行内部电源切割，提供数十个状态记录缓存器作为系统重启时的初始状态参考源。此模式的优点是更低的静态电流，通常仅需 100nA ~ 500nA，其缺点是并非所有的应用都可以忍受 RAM 数据丢失及系统重启。

## 电源系统的考虑

在多电源系统的应用上，必须考虑低功耗 MCU 的内部电源规划或自动切换，以下以市电/备用电池双电源系统及内建 USB 接口，但平常由电池供电的行动装置来举例说明。

市电/备用电池双电源系统：MCU 平常由市电经由交直流转换电路供电，当市电断电时，经由连接在备用电源的独立供电管脚进行供电，同时在 MCU 内部进行电源切割，并提供一个可靠的备用电源自动切换开关，确保市电正常供电时备用电池不会持续被消耗。但仔细考虑，其实有两种状况可能发生，一种是备用电池仅供电给部分低功耗的外围电路，例如 32.768K 晶振、RTC 时钟电路、数据备份寄存器等。当市电来时 MCU 将重新启动。另外一种状况是当市电断电时，有可能 MCU 及部分外围电路会被唤醒工作，然后再次进入待机模式。智能型电表就是此类应用的典型代表。在此种应用中，备用电池需要供电给整颗 MCU，所以电源自动切换开关必须能承受更高的电流，相对成本也较高。

内建 USB 接口行动装置：此类装置平时由两节电池供电或锂电池供电，工作电压可能为 2.2V 到 3V，当连接到 USB 时，USB 接口转由 VBUS 供电。此类低功耗 MCU 如果没有内建 5V 转 3V 的 USB 接口 LDO 将会产生下列问题，当连接 USB 时必须由外挂的 LDO 将 USB VBUS 的 5V 电源转换为 3V 电源同时

提供给 MCU VDD 及 USB 接口电路,但又必须避免 LDO 输出的 3V 电源与脱机操作时的电池电源发生冲突,将会需要外加电源管理电路,增加系统成本及复杂度。

### 丰富的唤醒机制及快速唤醒时间

有许多的系统应用场合,需要由外部的单一讯号、键盘或甚至串行通讯信号来激发 MCU 启动整体系统的运作。在未被激发的时候,微控器或甚至大部分的整机需要处于最低耗电的待机状态,以延长电池的寿命。能够在各式需求下被唤醒,也成为微控器的重要特征。MCU 能拥有各式各样的唤醒方式,包括各 I/O 可作为激发唤醒的通道,或是由 I2C、UART、SPI 的信道作为被外界组件触发唤醒,或使用内、外部的超低耗电时钟源,透过 Timer 来计时唤醒。诸多的唤醒机制,只要运用得当,并配合微控器的低耗电操作切换模式,可以使 MCU 几乎时时处于极低功耗的状况。

配有快速、高效率内核的 MCU,可以在每次唤醒的当下短暂时间里,完成应有的运作与反应,并再次进入深层的低待机模式,以此达到平均耗能下降的目的。但是,如果唤醒后开始执行微指令的时间因为某些因素而拖延的很长,将会使降低总体耗电的目标大打折扣,甚至达不到系统反应的要求。因此,有些 MCU,配合起振时间的改进,逻辑设计的配合,使得唤醒后执行指令的时间至少降到数个微秒之内。

### 低功耗模拟外围及内存

低功耗 MCU 在运行时除了 CPU 内核及被致能的数字外围电路在工作外,越来越多被整合到内部的模拟外围电路也是耗电的主要来源。以最简单的 while (1); 执行序来分析运行功耗,共包含下列耗电来源: CPU 内核、时钟振荡器、嵌入式闪存内存、及 LDO 本身的消耗电流。代入以下典型值数据将会更清楚显示各个部分对耗电的影响:

运行频率 12MHz, MCU 电压 3V, LDO 输出 1.8V 供给 CPU 内核、内存及其他数字电路

低功耗 CortexTM-M0 内核: 600  $\mu$  A

嵌入式闪存内存: 1.5 mA

低功耗 12MHz 晶震电路: 230  $\mu$  A

LDO 本身的静态消耗电流: 70  $\mu$  A

总和 = 0.6 + 2 + 0.23 + 0.07 = 2.4 mA, 平均功耗约 200 $\mu$  A/MHz

其中耗电比例最高的是嵌入式闪存内存。如果要运行在更高频率,通常会启动内建的 PLL 提供更高频率的时钟源,在 1.8V 供电的典型 PLL, 12MHz 输入输

出 48 MHz 工作电流约为 1 ~ 2mA, 如果不能有效降低 PLL 耗电, 对高频工作的低功耗 MCU 将是一大电流负担。

LDO 的最低静态功耗、32.768 kHz 晶振电路、BOD 及 TN LCD 驱动电路的工作电流, 都会大大影响到待机或 RTC 模式的功耗指针。以低功耗应用的热能表为例, RTC 加 LCD 显示的功耗要求在 3V/8 $\mu$ A 以下, 这代表可以预估分配给下列电路的电流预算为: LDO 静态功耗 0.5 $\mu$ A + 32.768 kHz 晶振及 RTC 电路 1 $\mu$ A + BOD 1 $\mu$ A + TN LCD 驱动 4 $\mu$ A + LCD 玻璃 1 $\mu$ A + 所有数字电路及模拟外围漏电流 0.5 $\mu$ A. 这些模拟外围除了低耗电要求, 同时必须兼具要求批量生产及温度变化时的一致性, 这对模拟设计人员将是一大挑战。

快速唤醒这个性能指针也会影响到下列模拟外围的稳定时间。当 MCU 从低功耗的待机模式唤醒时, 首先要将 LDO 快速切换到高供电模式, 启动内部高速 RC 震荡器, 使能嵌入式闪存及 CPU, 以上所有电路的稳定时间总和必须在数个微秒内完成, 才能符合快速唤醒的需求。

另外一个容易被忽略的设计是外围电路启动电流, 因为相当多的便携设备采用 CR2032 小型锂电池, 瞬间推动力仅有数 mA, 尤其使用一段时间瞬间推动力会更低, 当 MCU 被唤醒时果外围电路启动电流总和太大时, 将会导致 CR2032 输出电压骤降而导致 MCU 重置 (Reset) 或工作不正常。为了避免此问题, 除了降低外围电路的启动电流, 另一种方法是分时分段启动外围电路, 不要集中开启太多耗电的电路。

### 平均功耗计算范例

为了让读者更具体了解平均功耗的计算, 以新唐科技的低功耗 32 位 MCU Nano 系列及血糖计应用为例, 进行使用年限的预估。新唐的 Nano 系列低功耗 32 位 MCU 的 CPU 内核为 CortexTM-M0, 具有 200uA/MHz 低运行功耗、待机电流仅需 1uA、7uS 快速唤醒、多重时钟讯号来源及多种工作模式, 多达 128 KB Flash、16K SRAM 及 12 位 ADC、12 位 DAC、SPI、I2C、I2S、UART、LCD、Touch Key 等丰富外围, 符合低功耗、高性能 MCU 应用需求。

此血糖计范例采用 CR2032 230 mAh 电池, 使用方式、运行功耗及静态功耗如下表所示。

操作模式	参数	单位
每天使用次数	6	次
每次量测时间	0.25	分钟
每次显示时间	2	分钟
MCU 运行耗电流 (12MHz)	3	mA
外部量测电路耗电流	3	mA
待机 (含 RTC) 耗电流	2.5	$\mu$ A
LCD 耗电流	8	$\mu$ A
CR2032 自放电	0.25	$\mu$ A

使用年限的计算方式请参考下表。量测时间比例、显示时间比例及待机时间比例可由上表求得。例如，量测时间比例为“6 次 x 0.25 分钟 / (60 x 24) 分钟 = 0.1%”。其余时间比例依此类推。量测平均电流为“量测时间比例 x (MCU 运行耗电流 + 外部量测电路耗电流 + 待机 (含 RTC) 耗电流 + LCD 耗电流 + CR2032 自放电)”。显示平均电流为“显示时间比例 x (待机 (含 RTC) 耗电流 + LCD 耗电流 + CR2032 自放电)”。待机平均电流为“待机时间比例 x (待机 (含 RTC) 耗电流 + CR2032 自放电)”。最后计算出使用年限约为 2.77 年。由于待机时间比例高达 99%，故血糖计应用待机电流为延长使用年限最重要的参数。

功耗預算	使用比例	單位
量測時間比例	0.10%	-
顯示時間比例	0.83%	-
待機時間比例	99.06%	-
量測平均電流 (含 LCD 顯示)	6.26	μA
顯示平均電流	0.09	μA
待機平均電流 (不含 LCD 顯示)	2.72	μA
平均電流總和	9.08	μA
CR2032 總電量	220	mAH
CR2032 電壓	3	V
平均消耗功率	0.027	mW
CR2032 總功率	660	mW
使用年限	2.77	年

## 结论

低功耗 MCU 设计是一个需要多面向考虑的复杂工作，本文仅阐述基本设计理念。开发低功耗 MCU 产品时，不只要挑战电路设计的高难度，更要由客户应用的角度考虑性价比，功能最强的不一定是最好的。往往性价比最适合的才能在市场上取得成功。由于智能电网、物联网、远程控制、自动化管理等低功耗高效能应用需求量持续增加，在可以预见的未来，32 位低功耗 MCU 将逐渐取代 8/16 位低功耗 MCU，成为市场主流。