AN2857
应用笔记

STM8S 系列 MCU 功耗管理

介绍

本应用笔记面向系统设计人员，提供了STM8S系列产品各种低功耗模式的硬件实现概况；示范了如何在这些低功耗模式下使用STM8S产品；描述了测量功耗和(从低功耗模式)唤醒时间的方法，并给出了测量结果。

本应用笔记提供了测量STM8S不同模式功耗和唤醒时间的固件例程。

译注：

本译文的英文版下载地址为：

http://www.st.com/stonline/products/literature/an/15241.pdf

示例程序包下载地址：

http://www.st.com/stonline/products/support/micro/files/an2857.zip

目录

1 影响功耗的因素

在互补金属氧化物半导体(CMOS)数字逻辑电路中，功耗为下列因子之和：

- 静态功耗(主要由晶体管的偏置电流和漏电流产生)
- 动态功耗，取决于电源电压和工作时钟频率，按照下面公式：动态功耗 = CV^2f，其中
  - C为CMOS负载电容
  - V为电源电压
  - f为时钟频率

在CMOS逻辑电路以一定时钟频率运行时，静态功耗与动态功耗相比是可以忽略的。但在一些低功耗模式下，时钟不再运行，此时静态功耗是主要的功耗源。

因此，功耗主要取决于：

- **微控制器单元(MCU)的芯片面积**：所采用的工艺，晶体管的数量，片上集成和使用的模拟功能/外设。
- **MCU电源电压**：CMOS逻辑电路中消耗的电流与电源电压的平方成正比。因此，可以通过降低供电电压来降低功耗。
- **时钟频率**：在不要求进行高速处理的应用中，降低时钟频率可以降低功耗。
- **激活的外设数目或使用的MCU功能数目**(CSS、BOR、PVD……)：激活的外设数目越多，或使用的MCU功能数目越多，则功耗越大。
- **工作模式**：功耗会随着应用所处的不同功耗模式而改变(CPU开启/关闭，晶振开启/关闭……)。

对于由电池供电的系统来说，功耗是非常重要的指标。通常，要求系统的平均功耗小于某个目标值来保证一个适当的电池续航时间。这意味着系统可以在短时间内功耗较大，而把平均功耗维持在目标值以下。

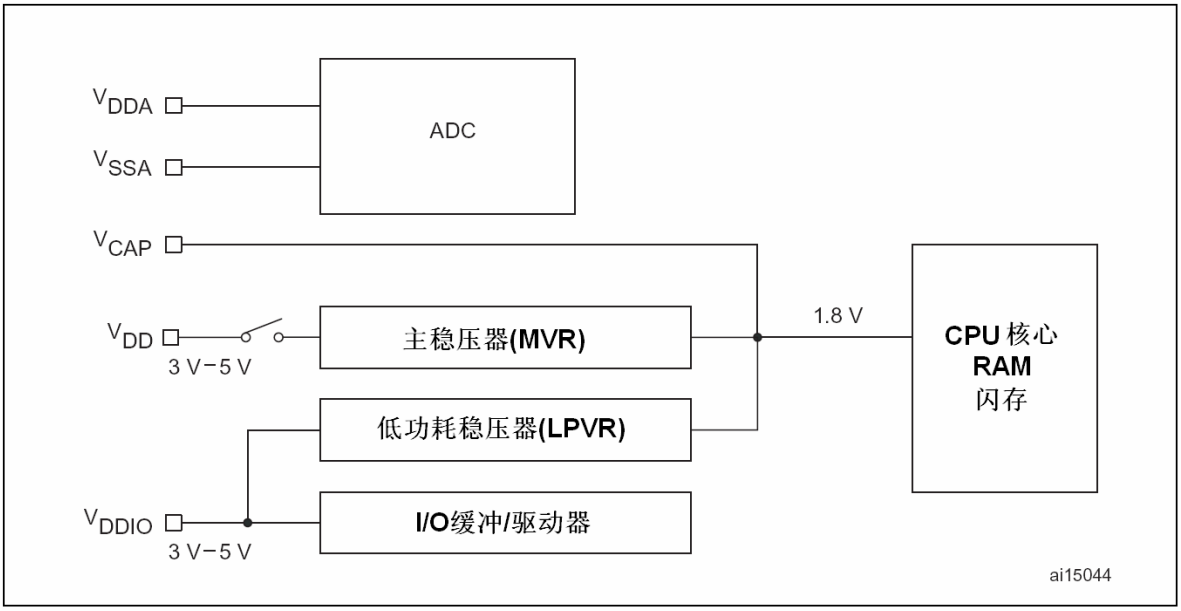
2 电源系统

2.1 内部电源结构

STM8S产品要求的工作电压范围(V_{DD})为3V到5.5V。芯片中嵌入了2个稳压器：主稳压器(MVR)和低功耗稳压器(LPVR)，在不同的功能模式下(见图1)对内部数字电路提供1.8V的电压。

V_{CAP}管脚用来对内部1.8V电压进行去耦。在该管脚和地之间应当连接一个容值至少为470nF的电容。绝对不允许用外部稳压器通过该管脚提供1.8V电压。访问ST网站www.st.com并参阅STM8S参考手册(RM0016，译注：此处英文版误为RM0009)获取更多相关细节。

图1　电源系统概览



2.2 模拟电源

STM8S产品的模数转换器(ADC)由独立的电源供电。数字和模拟电源应当用电容去耦，请参阅STM8S参考手册获取更多去耦电容相关细节。

2.3 IO电源

IO端口有专用的电源管脚，应当使用推荐的电容进行合适的去耦，请参阅STM8S参考手册获取更多去耦电容相关细节。

2.4 稳压器

在复位以后，主稳压器MVR提供1.8V电压给MCU内部数字电路。根据不同的功能模式，可以关闭MVR，改由低功耗稳压器LPVR提供1.8V电压。

- 在运行(RUN)和等待(WAIT)模式，只有MVR提供1.8V电压。在运行模式下不能使用LPVR。
- 在活跃停机(Active HALT)模式的停止阶段，MVR和LPVR都可以提供1.8V电压。用户可以选择使用哪个稳压器。
- 在停机(HALT)模式，自动启用LPVR，不能使用MVR。

3 时钟管理

3.1 时钟系统概述

可以使用4种不同的时钟源来驱动主时钟

- 来自晶体振荡器的1-24MHz外部高速时钟(HSE crystal)
- 用户提供的高达24MHz的外部高速时钟(HSE user-external)
- 16MHz内部高速RC振荡器(HSI)
- 128kHz的内部低速RC振荡器(LSI)

每个外设都可以在不使用的时候，单独地打开或者关闭它的时钟来优化系统功耗。该功能由外设门控时钟 (PCG)功能来实现。更多细节请参阅STM8S参考手册“时钟控制”的相关章节。

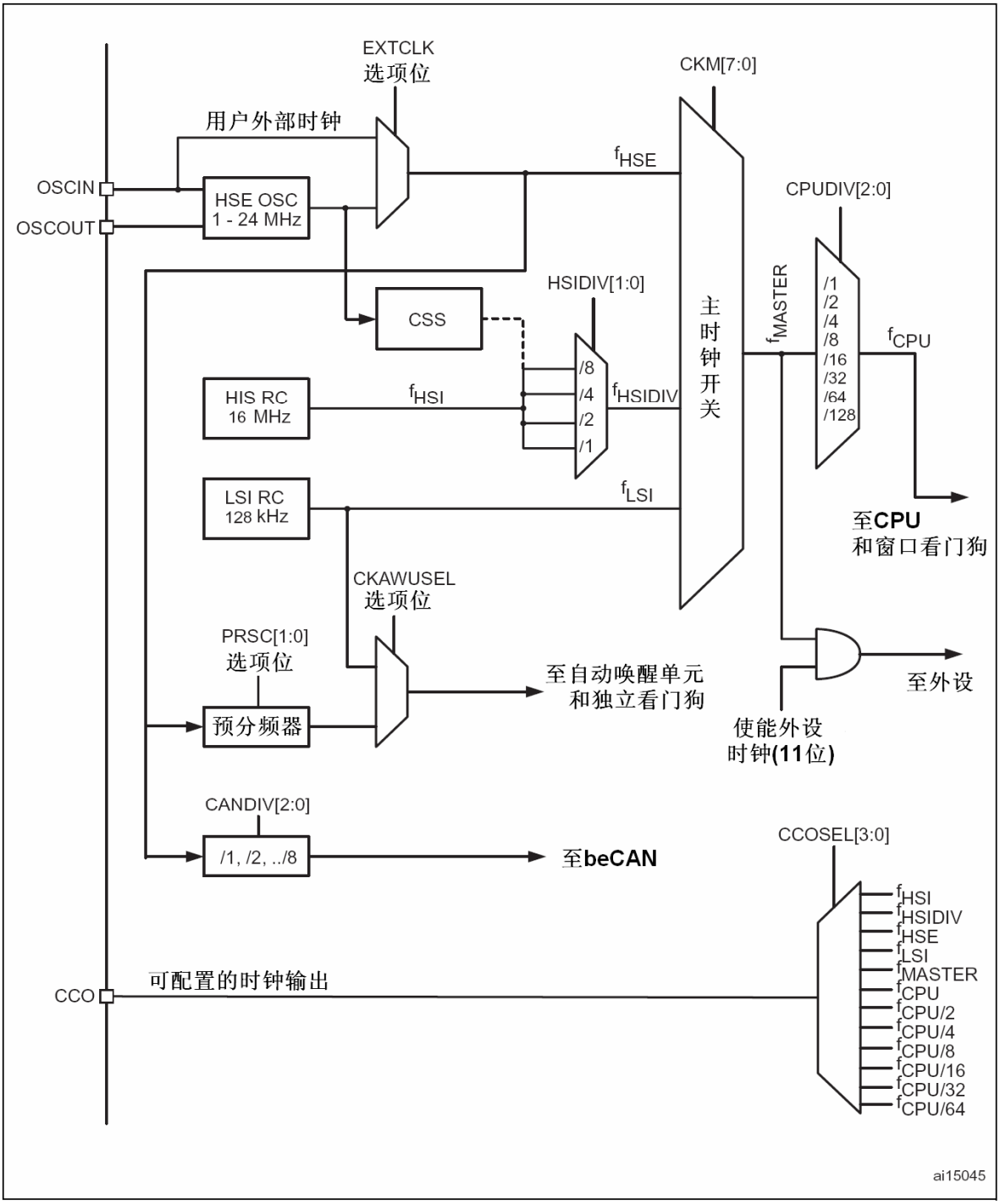
表1总结了各个时钟源的特点。STM8S提供完整的选择来应对不同用户对时钟在成本，精度和功耗方面的不同要求。

表1　时钟源比较

	HSE Crystal	HSE external	HSI	LSI
成本	低～中	免费～高	免费	免费
精度	取决于晶体	取决于外部时钟	±1%(1)	±2.5%(1)
功耗(1)	高	中	低	非常低
其他信息	共振器或晶体	现成时钟～非常复杂时钟系统	-	-

1. 由意法半导体在出厂时于T_A = 25 ℃校准。

图2　　时钟树



主时钟源(f_{MASTER})为内核和外设提供时钟，只读寄存器CLK_CMSR包含了当前主时钟源的选择信息。通过设置可写寄存器CLK_SWR可以选择"下一个"主时钟，在新的时钟源生效(细节见下文)以后，寄存器CLK_SWR的内容复制给寄存器CLK_CMSR。表2列出了时钟选择的数值。

复位以后，默认的时钟为HSI/8，随后用户可以把时钟切换到不同时钟源和不同频率：

- 编写寄存器CLK_CKDIVR的HSIDIV[1:0]位来选择16MHz内部RC振荡器的预分频数(/1、/2、/4或/8)。
- 变换主时钟为HSE或者LSI。更多时钟切换机制的相关细节，请参考STM8S参考手册。

当使用自动切换模式时，在关闭当前时钟源前，应当确保内核不再以当前时钟运行。就是说在标志位SWIF置'1'以后才能关闭当前的时钟。如果用户在硬件设置SWIF标志位为'1'之前就试图关闭时钟，由于MCU仍然基于当前时钟运行，该时钟是不会被关闭的。

这样的时钟切换也可以和等待模式结合使用(例如HSE外晶振作为新的主时钟源)，在切换时钟前进入等待模式，由时钟切换执行中断将MCU从等待模式唤醒，这样MCU唤醒后就可以和新的时钟同步。

时钟切换也可以用在常规或者中断路径的开头或者结尾，来加速代码执行(例如，主时钟是LSI，但是部分代码需要快速执行，可以切换到HSI来运行这些代码)。

注意： *在产品通过外部时钟以高于16MHz的频率运行时，必须设置选择项字节中的等待状态(wait state)位'1'，对闪存存储器访问时会增加一个周期的等待状态。*

表2　时钟选择表

寄存器值	主时钟源(CLK_CMSR)	下一主时钟(CLK_SWR)
E1h	HSI	HSI
D2h	LSI	LSI
B4h	HSE	HSE
其他	复位	保持当前主时钟

3.2 时钟设置和功耗管理

除了可以灵活选择时钟源，还有一些其他时钟设置可以用来优化产品的功耗：

- PCG：每一个外设的时钟都可以通过设置寄存器CLK_PCKENRx来打开或者关闭。
- CPU时钟分频数为1到128(寄存器CLK_CKDIV的CPUDIV[2:0]位)：可以在不影响外设时钟的情况下，降低CPU的时钟频率。

4 运行模式和低功耗模式

在复位后，微控制器默认工作在运行模式。按照寄存器HSIDIV的复位值，默认的CPU时钟频率为HSI的16MHz除以8。

在不需要CPU运行，例如等待某外部事件时，有好几个低功耗模式可用来节省能耗。用户应当在低功耗、短启动时间、适当的外设功能和可用的唤醒源这几个因素间，折衷选择使用哪一种低功耗模式。

STM8S产品具有3种主要的低功耗模式：

- 等待模式(CPU停止，外设保持工作)
- 活跃停机(Active HALT)模式(CPU停止，如果使能AWU(自动唤醒)和IWDG(独立看门狗)，则它们保持工作，其余外设停止)
- 停机(HALT)模式(一切工作停止)

在运行和等待模式下，还可以采取以下手段来降低功耗：

- 降低系统时钟频率
- 关闭不使用外设的时钟

表3总结了STM8S产品的所有功能模式。该表给出了不同工作模式下MCU各部分工作情况的概述。不同模式组合可以满足不同应用对功耗和唤醒时间的不同要求。

表3　功能模式

模式	稳压器	闪存	振荡器	CPU	外设	进入该模式方法	触发唤醒事件
运行(RUN)	MVR	开启	开启	开启	开启	-	-
等待(WAIT)	MVR	开启	开启	关闭	开启	执行WFI(1)指令	所有内部或外部中断和复位
活跃停机(Active HALT)	MVR	开启	关闭，除了LSI或LSE	关闭	AWU和IWDG(如果打开)	在执行HALT指令后打开AWU	AWU或者外部中断和复位
		关闭					
	LPVR	开启	关闭除了LSI				
		关闭					
停机(HALT)	LPVR	开启	关闭	关闭	关闭	执行HALT指令	外部中断和复位
		关闭					

1. WFI = Wait for interrupt 等待中断

注意：使用LPVR(低功耗稳压器)时，MVR(主稳压器)会自动关闭。

4.1 运行模式

运行模式是STM8S产品的默认工作模式，产品正常工作时使用该模式，功耗最大。使用PCG(外设时钟门控)和较低频率的时钟可以降低在这个模式下的功耗。

可以通过几个途径降低时钟频率。例如可以把LSI作为f_{MASTER}的时钟。设置选择项字节的LSI_EN位使MCU以LSI时钟运行。更多有关选择项字节的细节请参阅ST网站www.st.com上的STM8S数据手册。

4.2 等待模式

等待模式，或者更确切地说，等待中断模式，其设计目的是在不需要CPU运行的场合通过关闭内核来降低STM8S产品的功耗。当需要等待一个内部或者外部事件，再继续运行程序时，可以使用等待模式。与其在运行模式下等待事件，不如把MCU切换入等待模式。该模式还可以通过使用PCG(外设时钟门控)功能，和为主时钟选择较低频率的时钟源来进一步降低功耗。

4.2.1 进入等待模式

通过执行WFI汇编指令来进入等待模式，这使CPU停止工作，但保持外设和中断控制器会继续运行。

在进入等待模式时，自动使能中断。

4.2.2 退出等待模式

当发生一个内部或者外部的中断请求时，CPU从等待模式中唤醒并恢复工作。该模式提供了最短的唤醒时间。

下面例举了一些外设和模块，它们产生的中断能够使MCU退出等待模式：

- I^2C
- USART
- SPI
- CAN
- ADC
- AWU
- 外部中断
- 定时器
- 时钟控制器(执行时钟切换)

更多上述外设功能的细节请参阅STM8S参考手册。各外设在STM8S不同型号上可用情况的细节请参阅STM8S数据手册。

4.2.3 活动等级/低功耗模式控制

在超低功耗的应用中，MCU大部分时间是运行在等待/停机模式中，仅在需要执行特别任务的时候被唤醒(通过中断)。一些重复发生而又简单的任务可以直接在一个ISR(中断服务子程序)执行完成而不需要返回到主程序。为了处理这样情况，用户可以在进入等待模式之前把AL位置'1'，这样就不会在进入中断服务程序时进行相关寄存器的保存/恢复操作，从而减少了中断服务程序运行的时间。

在一些非常简单的应用中所有的操作都可以只在ISR中执行。对于一些更复杂的任务，如果中断子程序判断需要返回到主程序，可以简单地通过把AL置'0'来实现。

活动等级/低功耗模式控制功能只在等待模式下有效，不能用于活跃停机和停机模式下。

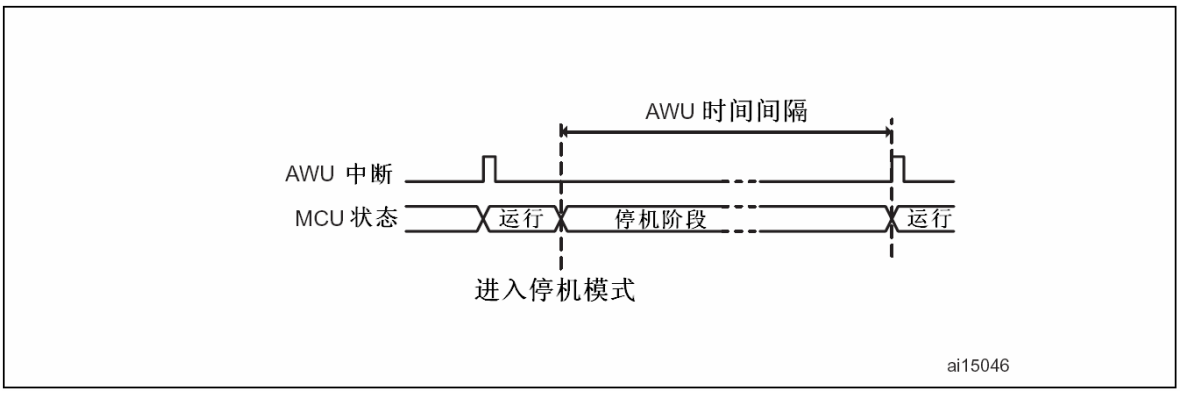
4.3 活跃停机模式

活跃停机模式可以认为是运行模式和停机模式之间的一种"混合"模式。它由2个阶段组成：

- 停机阶段：在此阶段，MCU处于停机模式，只有AWU单元在继续运行，如果使用LSI作为AWU的时钟源，那么LSI时钟也继续运行。
- 运行阶段：在此阶段，MCU处于运行模式。

在进入活跃停机模式时，AWU计数器开始计数，AWU中断会根据配置好的固定时间间隔周期性地唤醒CPU。一旦产品回到运行状态，AWU计数器停止计数。

图3　　活跃停机示意图



在活跃停机模式的停机阶段，用户可以选择使用哪个稳压器(主稳压器MVR或者低功耗稳压器LPVR)，也可以选择闪存的状态(工作模式或者掉电模式)。使用LPVR并将闪存设置成掉电模式可以降低功耗，但是会增加唤醒时间。

活跃停机模式可以有效地降低以MCU为核心系统的平均功耗。

表4　　活跃停机模式设置

稳压器	闪存状态	功耗(1)	唤醒时间(1)
MVR	开启	* * * *	*
	关闭	* * *	* *
LPVR	开启	* *	* * *
	关闭	*	* * * *

1. 星数越多，说明功耗越大，唤醒时间越长。

4.3.1 进入活跃停机模式

要进入活跃停机模式，应当设置和使能AWU，然后执行HALT指令。

注意：　如果在执行HALT模式前使能IWDG，那么MCU不会转入停机模式，而是转入活跃停机模式的停机阶段。这时如果没有使能AWU，MCU不会自动醒来，而只会由IWDG复位或者外部复位唤醒。

4.3.2 退出活跃停机模式

当发生AWU事件时，MCU回到运行模式。因此，AWU就是唤醒源。然而，在活跃停机阶段，任意停机模式下可用的唤醒源(见第4.4节)都可以把MCU唤醒。

快速时钟唤醒

对于活跃停机模式，MCU能较快地从中唤醒是很重要的。在2个低功耗阶段中间，MCU处于运行模式，减少这段时间既可以提高CPU的处理性能，也可以降低整体的平均功耗。

在一个唤醒事件发生以后，MCU会运行在进入活跃停机模式前选择的时钟。通常，由于振荡器稳定时间的影响，启动外部高速晶振需要的时间最长。STM8S提供了一种称为“快速时钟启动”的功能来减少启动时间。在唤醒事件后，MCU自动使用内部高速RC振荡器(HSI)，之后用户可以选择切换成另一种时钟，或者继续使用HSI时钟。

快速时钟唤醒功能默认为关闭。需要在进入低功耗模式之前，通过设置内部时钟寄存器(CLK_ICKR)的FHWU位为’1’打开这个功能。

4.3.3 停机阶段的稳压器和闪存设置

进入活跃停机模式时，主稳压器MVR是默认的稳压器，闪存处于工作模式。

可以通过设置内部时钟寄存器(CLK_ICKR)的SWUAH位来使用低功耗稳压器LPVR。设置闪存控制寄存器1(FLASH_CR1)的AHALT位可以把闪存置为掉电模式。

4.3.4 AWU单元

AWU(自动唤醒)单元是主动停机模式的核心。它可以以均匀的时间间隔产生中断，用来把MCU从停机模式唤醒。可以用2个时钟源对AWU提供时钟：

- LSI时钟
- 通过选择项字节选中HSE晶振，其频率除以预分频数(PRESC)得到约为128kHz的输入频率

AWU计数器只在停机阶段运行。一旦唤醒事件发生，MCU醒来，计数器就停止计数。当MCU再次进入停机阶段，计数器值归0。

以约128kHz时钟为输入，AWU可以提供范围从15.625微秒到30.720秒的唤醒时间间隔。

可以通过下面公式估算产品在活跃停机模式下的平均功耗(I_{TOT})。

公式1

$I_{TOT} = [t_{RUN} / (t_{RUN} + t_{AWU}) \times I_{RUN}] + [t_{AWU} / (t_{RUN} + t_{AWU}) \times I_{AHALT}]$　　(译注：此处原文有错)

t_{AWU}是处于停机阶段的时间，也是AWU的时间基数。

I_{RUN}是运行模式下的功耗电流，它受很多因素影响。可以按照表5和表6估计该参数的值。

I_{AHALT}是活跃停机模式的停机阶段的功耗电流。可以使用表8、表9和表10给出的数值。

t_{RUN}是在2个停机阶段中间，MCU处于运行阶段的时间。这个时间可以用下面公式计算

公式2

$t_{RUN} = t_{WKUP} + t_{CODE} + t_{SHDW}$

t_{WKUP}是从MCU醒来到执行第一条中断程序指令的时间。它取决于停机阶段的设置(所用的稳压器，闪存的状态)。表11、表12、表13和表14给出了一些该参数的典型值。

t_{CODE}是执行用户程序的时间。它取决于用户的代码。

t_{SHDW}是用来重新进入低功耗模式的时间。它取决于它取决于停机阶段的设置(所用的稳压器，闪存的状态)。这段时间也包括保存寄存器现场的时间。

4.4 停机模式

在停机模式，所有时钟停止工作，保留RAM和寄存器的值。在此模式下，关闭主稳压器MVR以降低功耗，只有低功耗稳压器LPVR工作。

4.4.1 进入停机模式

执行HALT指令使MCU进入停机模式。

4.4.2 退出停机模式

来自GPIO端口，或者来自外设的外部中断可以触发MCU从停机模式中唤醒。可以唤醒MCU的中断有：

- 外部中断(GPIO)
- CAN接收中断
- SPI传输结束
- I^2C中断(从地址匹配)
- 复位

4.4.3 在停机模式下闪存设置

进入停机模式后，闪存状态默认为掉电模式。要将闪存在停机模式下保持在工作状态，必须把闪存控制寄存器(FLASH_CR1)的HALT位置1。

闪存处于掉电状态时，可以降低功耗，但是增加唤醒时间。

5 功耗与唤醒事件的测量和结果

下文的测量及结果目的是突显不同低功耗模式对MCU功耗和唤醒时间的影响。

功耗的结果都是在25℃下测得的典型值。它们仅可用来参考。本应用笔记附带的zip文件包含了测量应用的软件。有关产品正式的功耗参数，请参考STM8S数据手册的电器特性章节。

5.1 功耗测量和结果

测量了运行模式和其他低功耗模式下MCU的功耗。在表5和表10中给出了测量结果。

5.1.1 测量设置

- V_{DDA}，V_{DDIO1}，V_{DDIO2}和V_{DD}统一连接到V_{DD}
- V_{SSA}，V_{SSIO1}，V_{SSIO2}和V_{SS}统一连接到V_{SS}
- 所有端口设置为输出低电平(关闭单线接口模块SWIM)
- 关闭所有外设(包括那些默认使能的外设)
- 关闭所有可能关闭的外设时钟(见STM8S参考手册的CLK_PCKENRx寄存器定义)

一旦设置完成，配置MCU进入第4章所示的各种功能模式。测量的对象为一个LQFP80封装的STM8S208MBT6微控制器。这是整个系列的最高配置型号，其功耗最高。

硬件环境

在V_{DD}，V_{DDIO}和V_{DDA}可以同时测量在运行模式和其他低功耗模式的功耗。硬件设置如图4：$I_{TOT} = I_{VDDIO} + I_{VDD} + I_{VDDA}$

图4　　电源设置

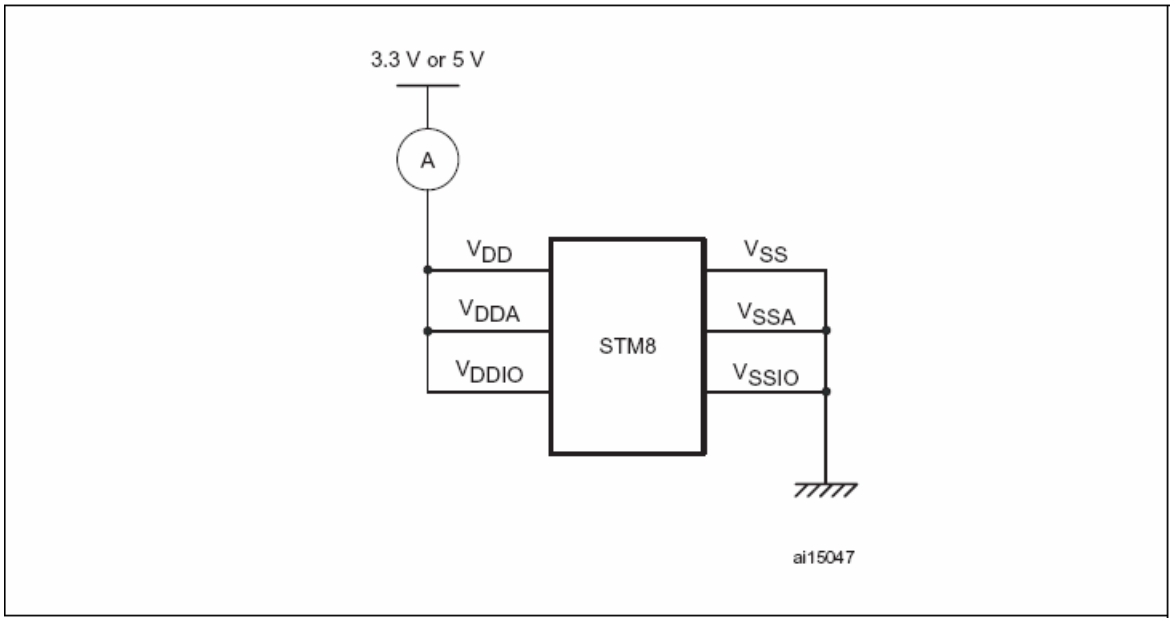
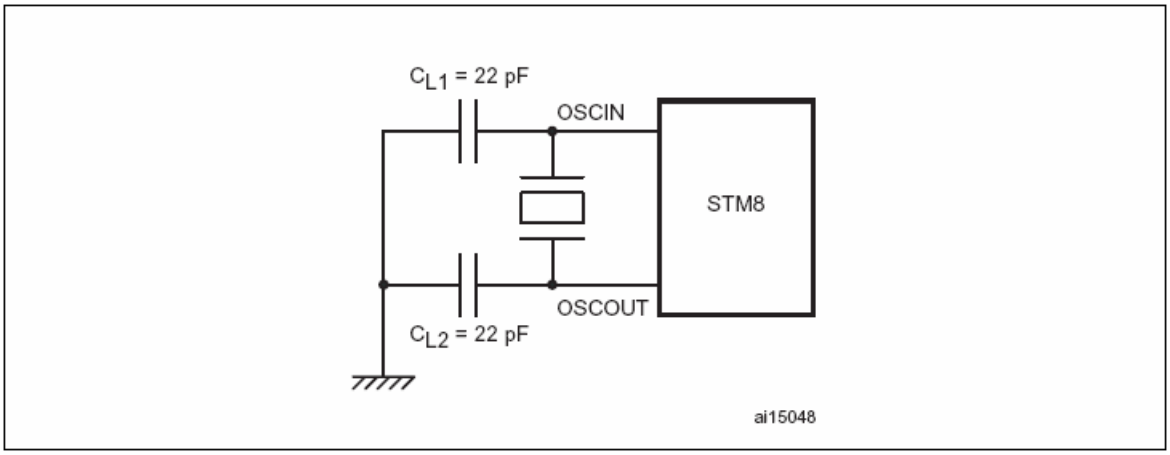


图4并没有显示所有电源管脚相关的去耦电容。为测量功耗，使用寄生串行电阻很小的陶瓷电容。这些电容上的漏电流很小，可以忽略不计。

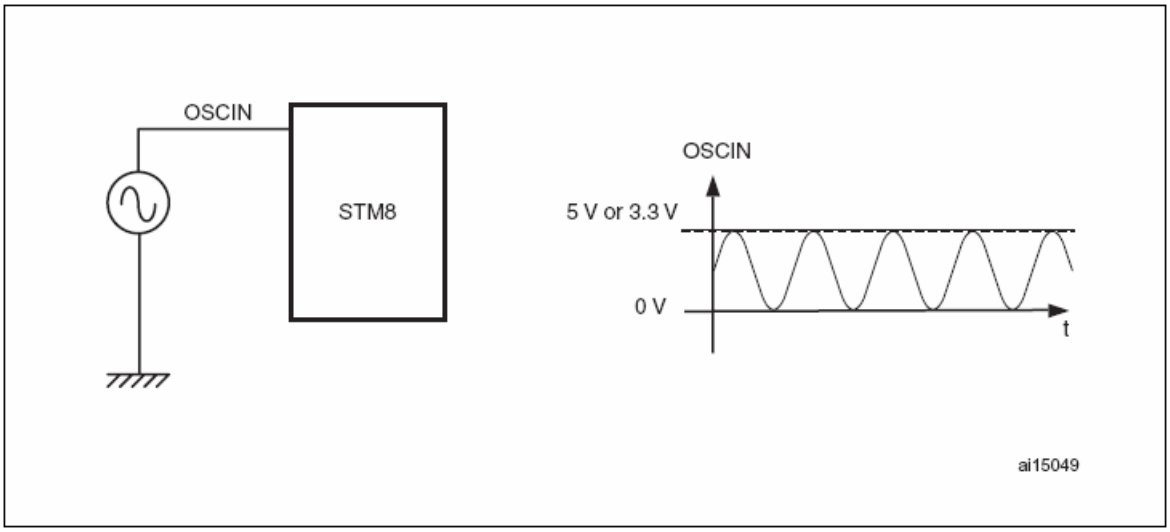
当使用外部晶振测量时，不使用内部的时钟分频。直接替换晶振来对应不同的频率。图5显示了晶体振荡器配置。

图5　　晶体振荡器配置



在使用外部时钟测量功耗时，使用以下配置：发生器产生正弦信号。根据电源电压不同，波形幅值或者为0~5V，或者为0~3.3V。由于没有用到OSCOUT管脚，该管脚设置与其他GPIO管脚相同(推挽输出V_{SS})。

图6　　外部时钟配置



通过设置外部时钟选择(EXTCLK)选择项位，选择使用外部晶振还是外部时钟。有关选择项字节的更多细节，请参阅STM8S数据手册。

运行模式下测量固件描述

MCU在运行模式下的功耗与多个因素有关，这些因素往往与应用程序关系密切更甚于运行模式本身。因此，很难给出功耗的典型值。例如，运行模式下的功耗与执行的代码和代码的位置有关(后者造成的功耗差异是由流水线和预取指缓存引起的)。

第5.1.2节给出了运行模式下，代码从闪存和RAM执行的功耗情况。代码首先对MCU进行设置(GPIO，PCG…)，然后进入一个无限循环，执行一系列不同指令100次。这么做是为了充分利用预取指缓存。

注意：　从闪存和RAM里执行的代码是相同的。

低功耗模式下测量固件描述

代码首先对MCU进行设置(GPIO、PCG、闪存模式、低功耗模式下使用的稳压器)。然后执行进入低功耗模式的指令，这样微控制器就切换进入相应的低功耗模式。在MCU确实进入低功耗模式之后，测量功耗。

有关固件设置阶段的细节，请参考本应用笔记附带的zip文件，该文件包含了相关的源代码。

5.1.2 运行模式下的功耗测量结果

表5　运行模式下的功耗测量结果，代码从FLASH执行

符号	参数	条件		V_{DD} = 3.3V	V_{DD} = 5V	单位
I_{TOT}	运行模式下功耗电流	HSE外晶振	$f_{CPU} = f_{MASTER}$ = 24 MHz	11.0	11.4	mA
		HSE外部时钟	$f_{CPU} = f_{MASTER}$ = 24 MHz	10.8	10.8	
		HSE外晶振	$f_{CPU} = f_{MASTER}$ = 16 MHz	8.4	9.0	
		HSE外部时钟	$f_{CPU} = f_{MASTER}$ = 16 MHz	8.2	8.2	
		HSI	$f_{CPU} = f_{MASTER}$ = 16 MHz	8.1	8.1	
		HSI	$f_{CPU} = f_{MASTER}$ /128 = 125 kHz	1.1	1.1	
		LSI	$f_{CPU} = f_{MASTER}$ = 128 kHz	0.55	0.55	

表6　运行模式下的功耗测量结果，代码从RAM执行

符号	参数	条件		V_{DD} = 3.3V	V_{DD} = 5V	单位
I_{TOT}	运行模式下功耗电流	HSE外晶振	$f_{CPU} = f_{MASTER}$ = 24 MHz	5.2	5.6	mA
		HSE外部时钟	$f_{CPU} = f_{MASTER}$ = 24 MHz	5.0	5.0	
		HSE外晶振	$f_{CPU} = f_{MASTER}$ = 16 MHz	3.7	4.1	
		HSE外部时钟	$f_{CPU} = f_{MASTER}$ = 16 MHz	3.5	3.5	
		HSI	$f_{CPU} = f_{MASTER}$ = 16 MHz	3.4	3.4	
		HSI	$f_{CPU} = f_{MASTER}$ /128 = 125 KHz	1.0	1.0	
		LSI	$f_{CPU} = f_{MASTER}$ = 128 KHz	0.47	0.47	

注意：　上表给出的数据与数据手册中给出的功耗数据有所不同，原因是测量用的代码有差异。本应用笔记中，从闪存执行和从RAM执行，执行的代码是相同的。

观察

- 功耗随着时钟频率的增加(f_{MASTER})而增加
- 在代码从RAM执行的时候，其代码的执行时间比从闪存执行相同代码要长。
- 在使用内部时钟的时候，功耗与电源电压无关，这是因为内部RC振荡器是由内部的1.8V稳压器供电的。
- 表5和表6没有体现出代码位于存储器不同位置的不同功耗。但是，如果循环代码的大小小于一个存储器块的大小，把循环代码放置在一个存储器块中可以降低功耗。因为这时只激活了一个存储器块。
- 表5和表6没有体现出执行不同代码的不同功耗。但是，运行模式下的功耗和执行的指令与指令的操作数有关。功耗相对低的指令有：INC A和NOP等。功耗相对高的指令有：LD(从闪存)和ADD A, #$01等。
- 外晶振的功耗大致为：
  - 5V：0.6 mA
  - 3.3V：0.3mA

5.1.3 等待模式下的功耗测量结果

表7　等待模式下的功耗测量结果

符号	参数		条件	V_{DD} = 3.3V	V_{DD} = 5V	单位
I_{TOT}	等待模式下功耗电流	HSE外晶振	$f_{CPU} = f_{MASTER}$ = 24 MHz	2.0	2.4	mA
		HSE外部时钟	$f_{CPU} = f_{MASTER}$ = 24 MHz	1.8	1.8	
		HSE外晶振	$f_{CPU} = f_{MASTER}$ = 16 MHz	1.6	2.0	
		HSE外部时钟	$f_{CPU} = f_{MASTER}$ = 16 MHz	1.4	1.4	
		HSI	$f_{CPU} = f_{MASTER}$ = 16 MHz	1.2	1.2	

		HSI	$f_{CPU} = f_{MASTER}$ /128 = 125 KHz	1.0	1.0	
		LSI	$f_{CPU} = f_{MASTER}$ = 128 KHz	0.50	0.50	

观察

- 功耗随着时钟频率的增加(f_{MASTER})而增加
- 在使用内部时钟的时候，功耗与电源电压无关，这是因为内部RC振荡器是由内部的1.8V稳压器供电的。
- 外晶振的功耗大致为：
  - 5V：0.6 mA
  - 3.3V：0.3mA

5.1.4 活跃停机模式下的功耗测量结果

在产品处于活跃停机模式的停机阶段时，进行测量。

表8 活跃停机模式下的功耗测量结果(MVR开LPVR关)

符号	参数	条件		V_{DD} = 3.3V	V_{DD} = 5V	单位
		时钟	闪存			
I_{TOT}	活跃停机模式下功耗电流(停机阶段)	HSE外晶振 24 MHz f_{AWU} = 24 MHz / PRSC = 128 kHz	开启	700	1260	μA
			关闭	640	1200	
		HSE外晶振 16 MHz f_{AWU} = 16 MHz / PRSC = 128 kHz	开启	600	1000	
			关闭	540	940	
		HSE外晶振 8 MHz f_{AWU} = 8MHz / PRSC = 128 kHz	开启	500	1040	
			关闭	440	980	
		HSE外晶振4 MHz f_{AWU} = 4 MHz / PRSC = 128 kHz	开启	460	970	
			关闭	400	910	
		LSI 128 KHz f_{AWU} = 128 kHz	开启	200	200	
			关闭	140	140	

表9 活跃停机模式下的功耗测量结果(MVR关LPVR开)

符号	参数	条件		V_{DD} = 3.3V	V_{DD} = 5V	单位
		时钟	闪存			
I_{TOT}	活跃停机模式下功耗电流(停机阶段)	LSI 128 kHz f_{AWU} = 128 kHz	开启	66	68	μA
			关闭	9	11	

5.1.5 停机模式下的功耗测量结果

表10 停机模式下的功耗测量结果(MVR关LPVR开)

符号	参数	条件	V_{DD} = 3.3V	V_{DD} = 5V	单位
I_{TOT}	停机模式下功耗电流	闪存在工作模式	61.5	63.5	μA
		闪存在掉电模式	4.5	6.5	

5.1.6 结论

功耗取决于：

- 电源电压
- 时钟频率

本章节的功耗测量结果显示，MCU的工作模式(运行、等待、活跃停机和停机)会影响功耗，采用恰当的低功耗模式可显著降低功耗。

在活跃停机和停机模式下，用户可以选择使用哪个稳压器(MVR或者LPVR)和使闪存处于哪个模式(工作模式或者掉电模式)。功耗测试方式参见第5.1节，相关外设的功耗估计如下：

- MVR：约135 µA
- 闪存：约60 µA
- AWU + LSI：约4 µA

5.2 唤醒时间的测量和结果

唤醒时间的测量在低功耗模式下进行，测量结果由表11和表14给出。

所测量的唤醒时间是指从中断事件发生开始，到中断的第一条指令执行之间的时间。图7解释了唤醒时间测量的具体情况。

注意：在STM8S数据手册中，唤醒时间的定义与本应用手册不同，它不包括取中断向量的时间。

5.2.1 测量设置

硬件环境

使用下面的硬件设置来进行唤醒时间的测量：微控制器的一个管脚用作中断输入管脚，另一个管脚作为输出管脚，同时在CLK_CCO管脚上输出CPU时钟。把这三个管脚的输出波形都显示在示波器上。

固件描述

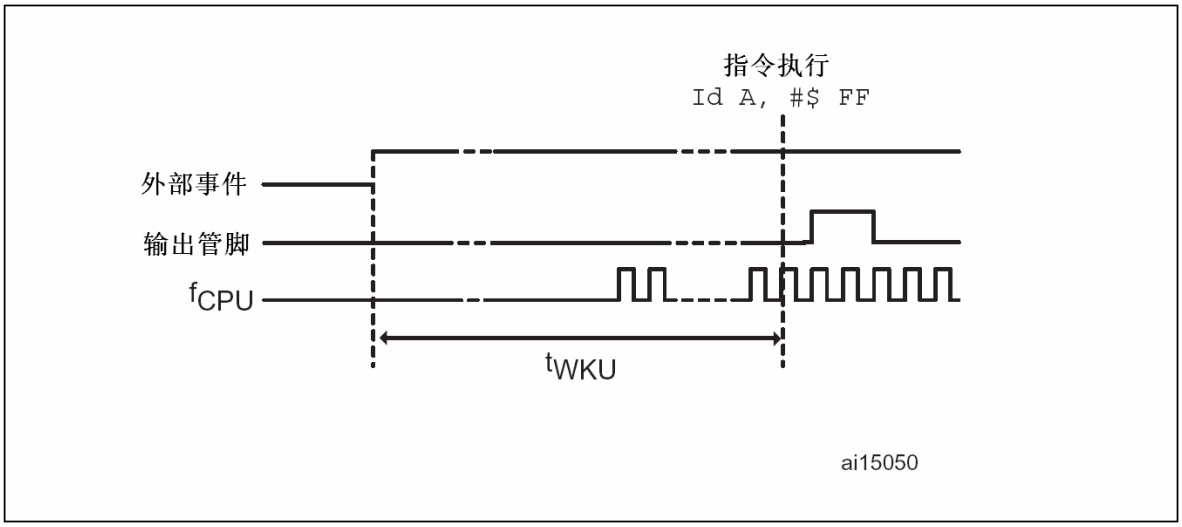
代码由MCU的设置开始。设置管脚(PB5)为中断输入，设置结束后，MCU进入低功耗模式。当触发外部中断时，MCU唤醒，并开始执行中断路径。中断会使一个管脚(PB1)翻转。

注意：本章节给出的测量结果考虑了中断延时，因为它们包括外部事件发生到执行中断路径的第一条指令之间的时间。

中断中执行的代码如下：

```
ld   A, #$02
ld   $5005, A
ld   A, #$00
ld   $5005, A
```

图7　唤醒时间测量示意图



5.2.2 唤醒时间测量结果

通常情况下，电源供电电压对唤醒时间没有影响。表11和表14中的符号(-)表示在2个不同的电源电压下，唤醒时间是相同的。

5.2.3 等待模式下唤醒时间测量结果

从等待模式唤醒的时间每次并不相同，原因是对唤醒事件的侦测是以CPU时钟频率进行采样的。因此其变化范围为一个CPU时钟周期，表11给出的是最大值。

表11　等待模式下唤醒时间测量结果

符号	参数		条件	V_{DD} = 3.3V(1)	V_{DD} = 5V	单位
$t_{WU(WFI)}$	等待模式下唤醒时间	HSE外晶振	$f_{CPU} = f_{MASTER}$ = 24 MHz	-	1.0	µs
		HSE外晶振	$f_{CPU} = f_{MASTER}$ = 16 MHz	-	0.75	
		HSE外晶振	$f_{CPU} = f_{MASTER}$ = 8 MHz	-	1.4	
		HSE外晶振	$f_{CPU} = f_{MASTER}$ = 4 MHz	-	2.7	
		HSI	$f_{CPU} = f_{MASTER}$ = 16 MHz	-	0.75	
		HSI	$f_{CPU} = f_{MASTER}$ /128 = 125 kHz	-	80	
		LSI	$f_{CPU} = f_{MASTER}$ = 128 kHz	-	94	

1. 符号(-)表示唤醒时间与V_{DD} = 5V时相同。

观察

- 在不附加闪存等待状态的情况下，从等待模式中唤醒的时间只和f_{MASTER}和f_{CPU}时钟频率有关。唤醒时间可以按照下面公式计算：

公式3

$t_{WU(WFI)min} = (2 \times 1 / f_{MASTER}) + (9 \times 1 / f_{CPU})$

公式4

$t_{WU(WFI)max} = (2 \times 1 / f_{MASTER}) + (10 \times 1 / f_{CPU})$

- 时钟为24MHz时唤醒时间比16MHz时长。这是因为在访问闪存时，插入了等待的时间。

5.2.4 活跃停机模式下的唤醒时间测量结果

对于活跃停机模式下的测量，总是用LSI作为AWU的时钟。在表12和表13中，条件/时钟一栏的意思是进入活跃停机模式前使用的时钟。

从活跃停机模式唤醒的时间每次并不相同，其原因是对唤醒事件有与等待模式相似的采样机制。唤醒时间的变化范围为1个AWU时钟周期(7.8125µs)。表12和表13给出的数据为最大值，包括了AWU时钟周期的采样。

表12　活跃停机模式下，唤醒时间测量结果(MVR)

符号	参数	条件		V_{DD} = 3.3V(1)	V_{DD} = 5V	单位
		时钟	其他			
$t_{WU(AH)}$	活跃停机模式下唤醒时间	HSI (16 MHz)	闪存开启	-	9	µs
			闪存关闭	-	11	
		HSE (16 MHz晶振)	快速时钟唤醒开启 闪存关闭	-	11	
			快速时钟唤醒关闭 闪存关闭	394	543	
		HSE (4 MHz晶振)	快速时钟唤醒开启 闪存关闭	-	11	
			快速时钟唤醒关闭 闪存关闭	1627	1274	

1. 符号(-)表示唤醒时间与V_{DD} = 5V时相同。

表13　活跃停机模式下，唤醒时间测量结果(LPVR)

符号	参数	条件		V_{DD} = 3.3V(1)	V_{DD} = 5V	单位
		时钟	其他			
$t_{WU(AH)}$	活跃停机模式下唤醒时间	HSI (16 MHz)	闪存开启	-	56	µs
			闪存关闭	-	58	

1. 符号(-)表示唤醒时间与V_{DD} = 5V时相同的。

观察

- 如果使能快速时钟唤醒，唤醒时间总是和HSI唤醒时间相同，因为此时MCU总是把HSI作为唤醒后的时钟。
- 如果唤醒后时钟是HSE晶振，唤醒时间会很长，而且与电源电压有关。这是因为振荡器达到稳态需要时间。可以通过下面手段避免过长的唤醒时间：
  - 使用快速时钟唤醒功能
  - 编写选择项字节HSECNT来设置一个较短的稳定时间。默认在使用HSE晶振的时候信号前会等待2048个振荡器时钟周期(上述测量也是基于2048个振荡器时钟周期这个默认值)。
- 表12没有给出条件为“HSE外晶振 / 闪存开启”选项下的唤醒时间，这是因为闪存的唤醒时间总是一样的。闪存的唤醒时间大致为2µs。
- MVR的唤醒时间大致为50µs。

5.2.5 停机模式下的唤醒时间测量结果

表14　停机模式下，唤醒时间测量结果(LPVR)

符号	参数	条件		V_{DD} = 3.3V	V_{DD} = 5V	单位
		时钟	其他			
$t_{WU(H)}$	停机模式下唤醒时间	HSI (16 MHz)	闪存开启	-	52	µs
			闪存关闭	-	54	

观察

- 闪存的唤醒时间大致为2µs(与活跃停机模式下相同)
- 停机模式下(用LPVR)唤醒时间比在活跃停机模式下要短(见表12)。因为表12的测量数据包括了1个AWU时钟周期的采样延时。

5.2.6 结论

整体唤醒时间为下列要素之和：

- 主稳压器(MVR)的唤醒时间(如果在低功耗模式下关闭它的话)。
- 振荡器的稳定时间(如果在低功耗模式下关闭它的话)。
- 闪存存储器的唤醒时间(如果在低功耗模式下将其置为掉电状态的话)。
- 唤醒触发事件的中断延时。

6 功耗管理要点

6.1 最小化功耗规则

- 关闭所有不使用的外设(除了USART、LINUSART和SWIM以外所有外设默认关闭)，并使用PCG功能(通过CLK_PCKENRx寄存器)来关闭不使用外设的时钟(时钟默认为打开)。更多相关细节请参阅第3.2节。
- 在运行模式下，如果一个循环的大小小于一个存储器块，循环代码必须位于同一个存储器块中。
- 所有不使用的管脚都应当设置为输出低电平。绝对不能设置任何不用管脚为悬空输入状态，这会导致不必要的高功耗。
- 如果需要某些外设持续工作和外部中断功能时，选择等待模式作为低功耗模式。
- 选择合适的V_{DD}值，V_{DD}值越高，功耗越高。
- 使用尽可能小的时钟频率，可以通过设置CPU的8个预分频数和4个HSI预分频数来得到适合应用程序的时钟频率。

6.2 为应用选择最佳的低功耗模式

- 由电池供电的应用，在大多数时间MCU处于睡眠模式：
  - 如果MCU由外部事件唤醒，且唤醒周期并不固定，而要求功耗尽可能低。这时，建议使用停机模式来尽可能延长电池续航时间。
  - 如果MCU的唤醒不依靠外部事件，而是依赖于某个不严格精准的周期，建议使用AWU和活跃停机模式。
- 由电池供电的应用，在大多数时间MCU处于唤醒模式：
  - 如果MCU需要执行一些周期性的工作，且没有需要持续运行的外设，建议使用活跃停机模式
  - 如果至少有一个外设要持续工作，且中断可以唤醒MCU，建议使用等待模式。
- 由固定电源供电的应用，但是功耗是重要指标
  - 如果需要MCU持续工作，建议使用运行模式，但是应当选择时钟预分频系数获得合适的时钟频率。