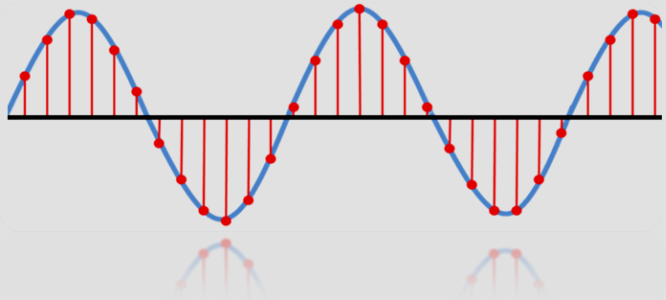
Stellaris® ARM® Cortex™ M4F 培训

外设概述

低功耗



模拟外设

USB

Stellaris® ARM® Cortex™ M4F 培训
低功耗

低功耗模式 & 休眠模块

图像：http://thumbs.dreamstime.com/thumblarge_457/1259665112fPx9gp.jpg

议程 – 功耗

- **Section 1**
  - 功耗概述

- **Section 2**
  - 如何基于**EK-LM4F232**评估板开始休眠模块开发
- **Section 3**
  - 练习

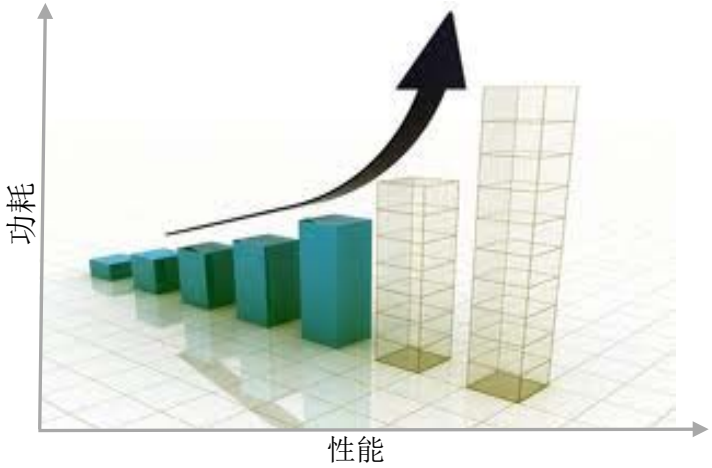
Section 1: 低功耗概述

- 功耗模式
- 休眠（**Hibernation**）模块：主要特点与应用
- 方框图与信号说明
- 功能模块
  - 实时时钟
  - 供电电源控制
  - 电池管理
- 实际应用考虑
- 利用驱动程序库 **API**简化应用设计

Section 1: 低功耗概述

- 功耗模式
- 休眠（Hibernation）模块：主要特点与应用
- 方框图与信号说明
- 功能模块
  - 实时时钟
  - 供电电源控制
  - 电池管理
- 实际应用考虑
- 利用驱动程序库 API简化应用设计

功耗模式



- **Stellaris M4F系列单片机**
  - 优化设计：更高性能，更低功耗
  - 适合更广泛的应用
- 在功耗（时钟速度）与性能 **(MIPS)** 之间需要进行权衡折衷
- 针对不同应用，为了优化在各种应用中的整体功耗指标，**Stellaris M4F** 系列 **MCU** 可工作于不同的模式：
  - 运行模式 **(RUN)**
  - 睡眠模式 **(Sleep)**
  - 深度睡眠模式 **(Deep Sleep)**
  - 休眠模式 **(Hibernation)**

运行模式（RUN ）

I_{DD_RUN}=32mA

– 运行模式下：

  • 处理器正常工作

  • 当前被启用的外设正常工作（即那些通过 RCGC 寄存器启用的外设）

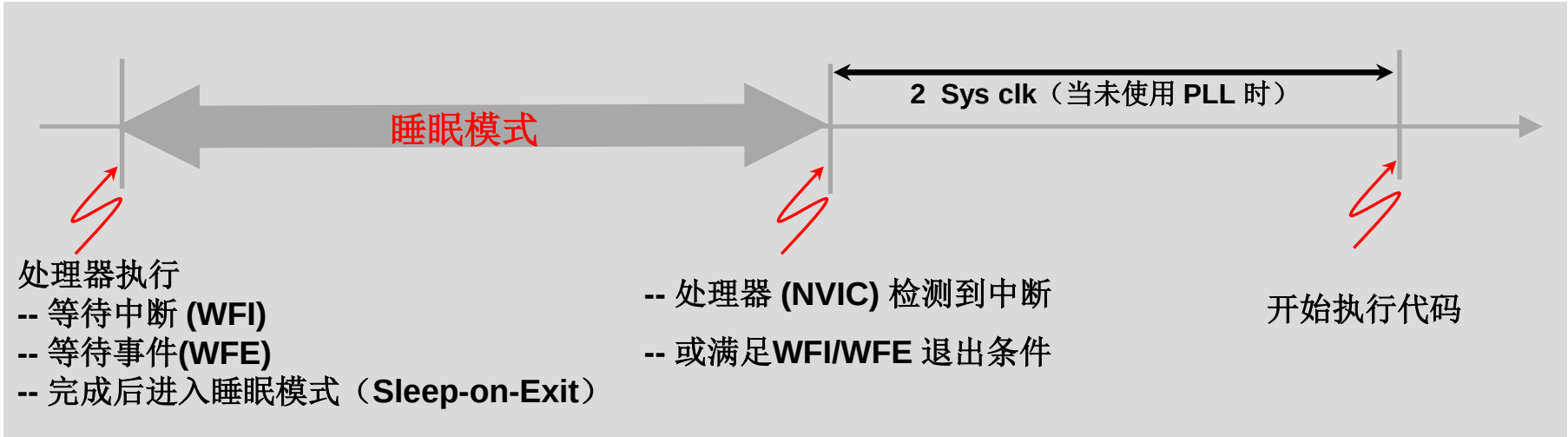
– MCU以最高性能执行代码

– 系统时钟可根据需要配置为任意可用时钟（包括 PLL）。

睡眠模式（Sleep）

I_{DD_SLEEP} = 10mA

- 睡眠模式：关断内核和存储器（Flash和SRAM）的时钟供给来降低整体功耗
- 被使能的外设继续以系统时钟正常工作
- 系统时钟的时钟源和频率不变（与运行模式时相同）
- 适用场合：
  - 处理器不需要执行代码程序
  - 且外设仍然需要以系统时钟速度工作

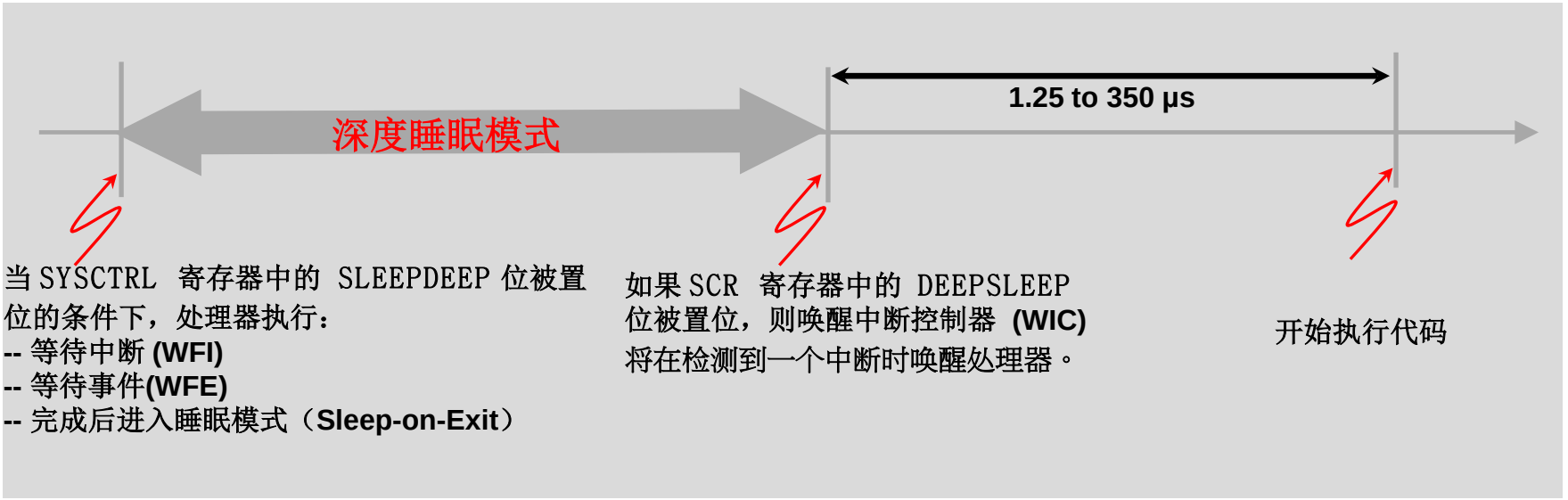


进入睡眠模式及从睡眠模式唤醒

深度睡眠模式 (Deep Sleep)

- 深度睡眠模式: 关断内核和存储器（**FLASH**和 **SRAM**）的时钟供给及以下措施进一步降低整体功耗：
  - 关断主振荡器电源（**MOSC，4～25MHz**）
  - 关断**PLL**电源（如果工作于运行模式）
  - 被使能的外设（由DCGC 寄存器使能）的时钟可由内部振荡器（**IOSC30kHz**）供给

- 适用场合：
  - 处理器不必运行代码程序
  - 且外设可以以低于运行模式时的时钟速度工作



进入深度睡眠模式及从深度睡眠模式唤醒

休眠模式（Hibernation）

I_{HIB}=1.6 μA

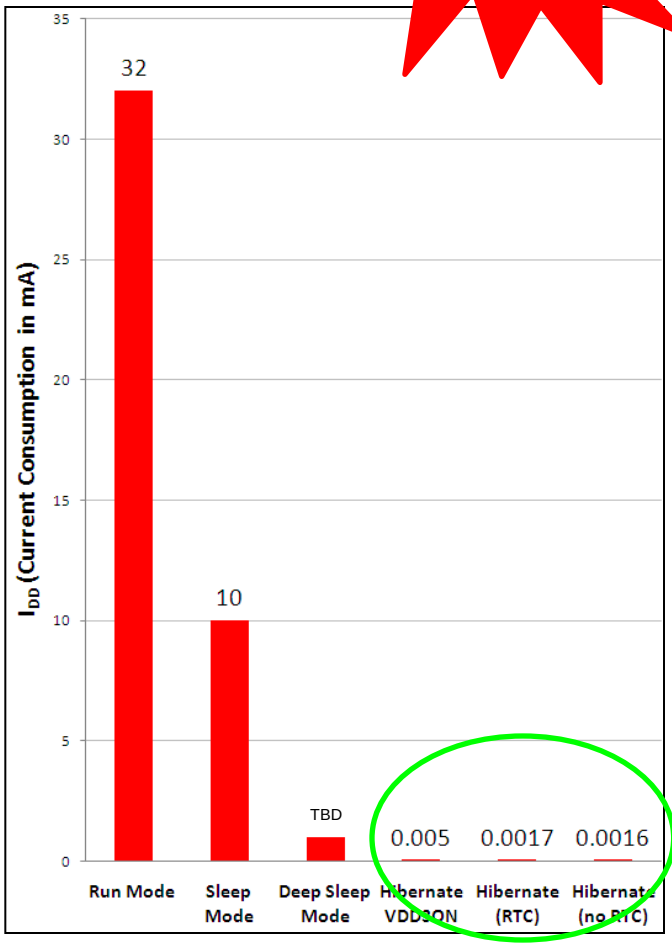
- 休眠模式是Stellaris LM4F系列单片机上的最低功耗模式
- 休眠模式下：
  - -- 完全切断内核与外设的供电
  - -- 只给Hibernate模块供电
- Hibernate硬件模块，具有以下功能：
  - -- 管理内核供电的取消和恢复
  - -- 可由外部电池或辅助电源单独供电
  - -- 允许在某个外部信号有效时或某个设定时刻恢复供电（RTC）
- 适用场合：

  在电池供电型/手持式应用中使用（此类应用中，系统可被置于最低功耗状态、同时保存某些状态信息）。

功耗模式比较

全新设计

模式 → / 参数 ↓	运行模式	睡眠模式	深度睡眠模式	冬眠 (VDD3ON)	冬眠 (RTC)	冬眠 (无 RTC)
I_{DD}	32 mA	10 mA	TBD	5 µA	1.7 µA	1.6 µA
V_{DD}	3.3 V	3.3 V	3.3 V	3.3 V	0 V	0 V
V_{BAT}	N.A.	N.A.	N.A.	3 V	3 V	3 V
系统时钟	40 MHz（PLL）	40 MHz（PLL）	30 kHz	Off	Off	Off
内核	供电	供电	供电	Off	Off	Off
	Clocked	Not Clocked	Not Clocked	Not Clocked	Not Clocked	Not Clocked
外设	On	Off	Off	Off	Off	Off
代码	while{1}	N.A.	N.A.	N.A.	N.A.	N.A.



不同功耗模式下的电流

议程

- 功耗模式
- 休眠（Hibernation）模块：主要特点与应用
- 方框图与信号说明
- 功能模块
  - 实时时钟
  - 供电电源控制
  - 电池管理
- 实际应用考虑
- 利用驱动程序库 API简化应用设计

主要特点

- 具有 15 位秒以下精度和可调功能的 32 位实时秒计数器（实时时钟）
- 具有专用引脚，可由外部信号唤醒
- V_{BAT} 供电可保证MCU处于休眠模式且 RTC 正常工作
- 休眠模式下仍可保证GPIO 引脚状态不变（VDD3ON 模式）
- 两种机制有效控制功耗
  - 系统功耗控制
  - 片上功耗控制
- 利用可选的低电量电池唤醒功能，检测电池电量低，并置标志产生中断
- 时钟源为外部32.768-kHz 晶体或振荡器
- 16 个 32 位由电池供电的存储单元，可在休眠模式下用于保存有用的 数据
- 可编程中断：
  - -- RTC匹配
  - -- 外部唤醒
  - -- 电池电量低

适合应用领域

便携式、电池供电型设备

无线与连通性

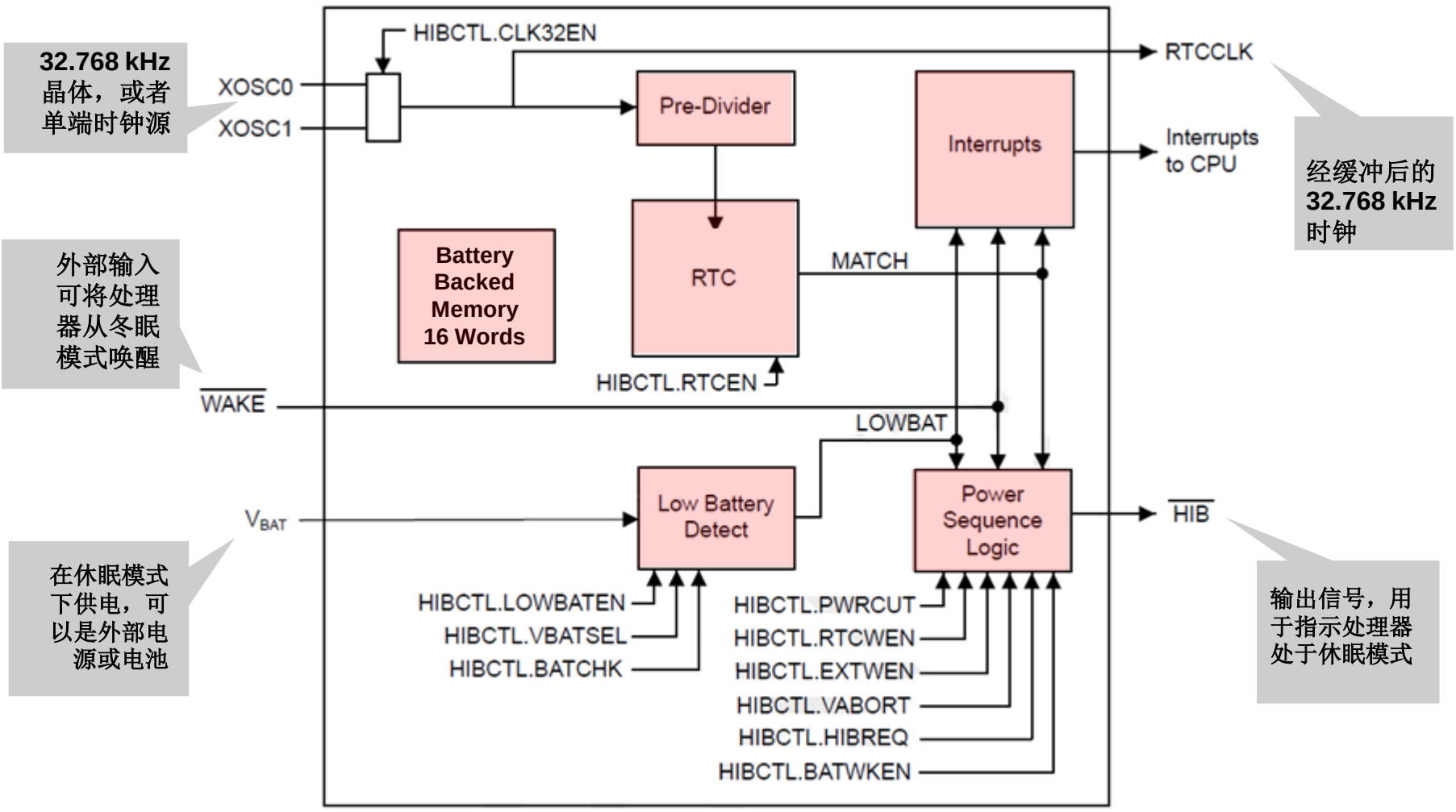
能源与计量

传感器应用

议程

- 功耗模式
- 休眠（Hibernation）模块：主要特点与应用
- 方框图与信号说明
- 功能模块
  - 实时时钟
  - 功耗控制
  - 电池管理
- 实际应用考虑
- 利用驱动程序库 API简化应用设计

方框图与信号说明



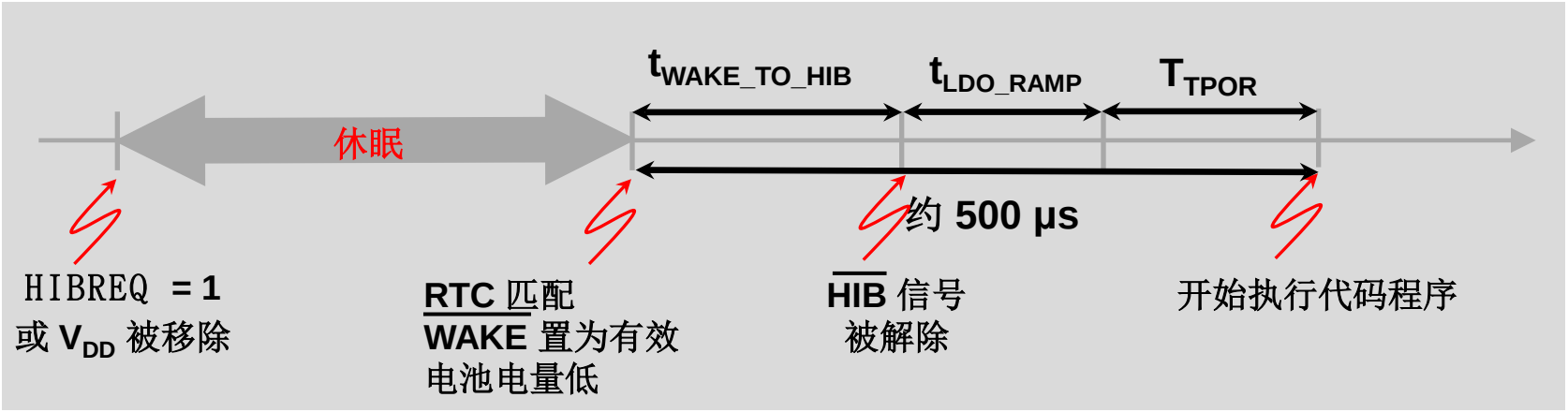
冬眠模块的功能方框图

议程

- 功耗模式
- 休眠（Hibernation）模块：主要特点与应用
- 方框图与信号说明
- 功能模块
  - 实时时钟
  - 供电电源控制
  - 电池管理
- 实际应用考虑
- 利用驱动程序库 **API**简化应用设计

功能描述

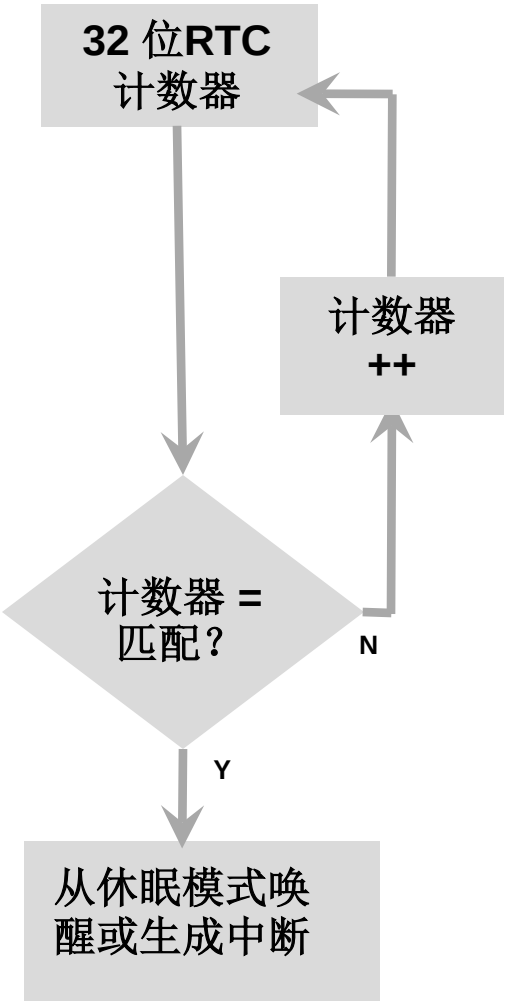
- **MCU**可通过以下方式进入休眠模式：
  - 当 HIBCTL 寄存器中的 HIBREQ 位被置位时，或者
  - 当 V_{DD} 被取消，而 V_{BAT}有效时（假如配置正确的话）。
- 当器件处于休眠模式时，$\overline{\text{HIB}}$ 信号被置为有效。
- **MCU**可通过以下方式从休眠模式唤醒：
  - 当 $\overline{\text{WAKE}}$ 引脚被置为有效时，或者
  - 当 **RTC** 匹配时，或者
  - 当检测到电池电量低时。
- 一旦被唤醒，$\overline{\text{HIB}}$ 信号即被解除，同时产生内部 **POR**。



进入冬眠模式及从冬眠模式唤醒

RTC：实时时钟

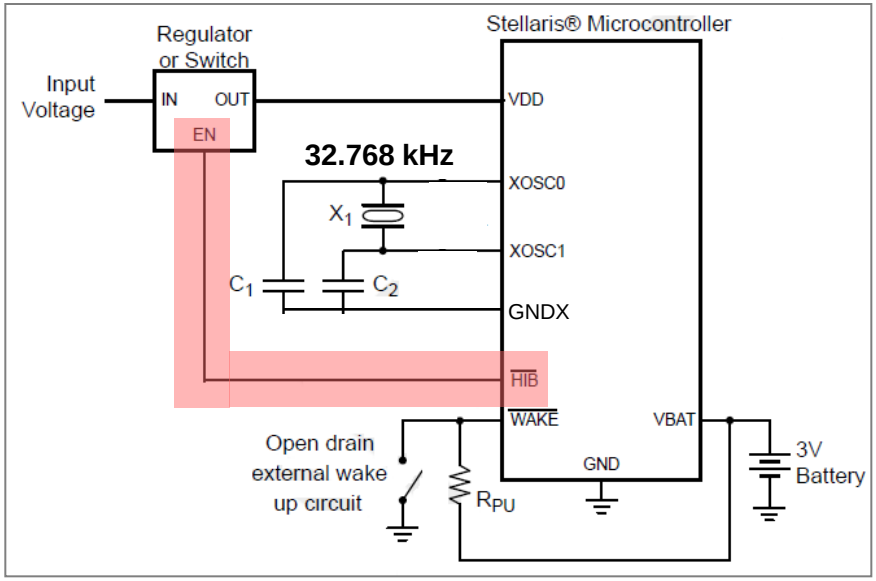
- 主要特点
  - RTC 时钟源：32.768kHz 休眠振荡器
  - 15 位秒以下精度，用户可调精度
  - 在 RTCCLK 引脚可提供经缓冲后的时钟
- RTC 包括一个 32 位秒计数器和一个 15 位辅助计数器。
  - 32 位计数器由 HIBRTCLD 寄存器配置
  - 当 32 位计数器被使能，则15 位计数器被自动清零
  - HIBRTCSS 和 HIBRTCC 寄存器能够存储一个 32 位匹配值和 15 位域值。
- 当休眠模块在下列场合中复位时，RTC 计数器复位：
  - 当 HIBCTL 寄存器中的 RTCEN 和 PINWEN 位被清零时，系统复位
  - 冷 POR，即：当 V_{DD} 和 V_{BAT} 均被移除时。
- 可根据需要使能RTC中断



供电电源控制

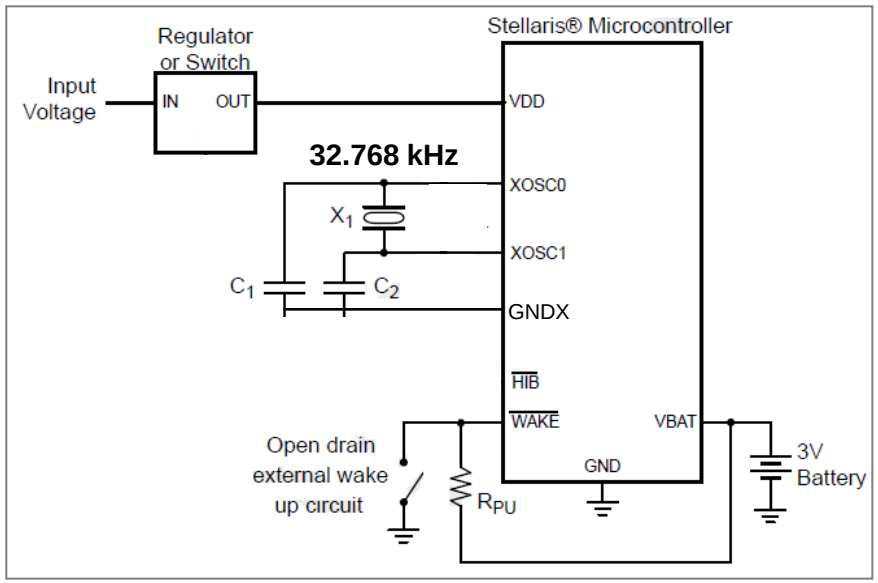
- 动态选择供电电源
  - 休眠模块的电源电压是 V_{DD} 和 V_{BAT} 中电压较高的那个。
- 用于电源控制的两种选择（由 HIBCTL 寄存器中的 VDD3ON 位配置）

利用一个控制信号控制 MCU 的供电，$\overline{\text{HIB}}$ 连接至一个外部 LDO 的 EN 信号。



采用一个外部 LDO 的电源结构

VDD3ON 模式：采用内部开关来控制 MCU 的供电，同时保持对 I/O 引脚的供电。



采用 VDD3ON 模式的电源结构*

*用于说明原理的简化示意图。详见数据表。

电池管理

- 电池电量低检测
  - 可通过配置在电池电压降至门限值以下时不进入休眠模式。
  - 监视外部电池的电压电平，当电压降至 V_{LOWBAT} （1.9V、2.1V、2.3V 或 2.5V）以下时及时地检测到。
  - V_{LOWBAT} 门限采用 HIBCTL 寄存器中的 VBATSEL 位进行配置。
  - 在休眠模式期间监测电池电压，通过配置使得当电池电压降至门限以下时从休眠模式唤醒（采用 HIBCTL 寄存器中的 BATWKEN 位）。
- 休眠模块由 V_{BAT} 和 V_{DD} 中电压较高的那个电源供电
- V_{BAT} （最小值 = 1.8V，最大值 = 3.6V）

议程

- 功耗模式
- 休眠（Hibernation）模块：主要特点与应用
- 方框图与信号说明
- 功能模块
  - 实时时钟
  - 供电电源控制
  - 电池管理
- 实际应用考虑
- 利用驱动程序库 **API**简化应用设计

实际应用考虑

- 寄存器写入
  - 在数据有效之前需要 2～3 个 32kHz 时钟周期进行同步。
  - 数据在写入完成位被置位之后有效，并可被查询或配置相应中断。
  - 读取操作以系统时钟为其时钟频率。
- 复位
  - 当启用外部引脚唤醒功能或 RTC功能时，系统复位被阻止。
- 进入休眠模式前的软件要求
  - 必须设定一个有效的唤醒信号源
  - 必须清除任何的低电池电量状况
  - 闪存/EEPROM 一定不得处于busy状态

议程

- 功耗模式
- 休眠（Hibernation）模块：主要特点与应用
- 方框图与信号说明
- 功能模块
  - 实时时钟
  - 供电电源控制
  - 电池管理
- 实际应用考虑
- **利用驱动程序库 API简化应用设计**

利用驱动程序库 API简化应用设计

StellarisWare®

- 在驱动程序库 (DriverLib) 中提供休眠模式 API，助您快速启动项目开发

  - void HibernateClockSelect (unsigned long ulClockInput)
  - void HibernateDataGet (unsigned long *pulData, unsigned long ulCount)
  - void HibernateDataSet (unsigned long *pulData, unsigned long ulCount)
  - void HibernateDisable (void)
  - void HibernateEnableExpClk (unsigned long ulHibClk)
  - void HibernateIntClear (unsigned long ulIntFlags)
  - void HibernateIntDisable (unsigned long ulIntFlags)
  - void HibernateIntEnable (unsigned long ulIntFlags)
  - void HibernateIntRegister (void (*pfnHandler)(void))
  - unsigned long HibernateIntStatus (tBoolean bMasked)
  - void HibernateIntUnregister (void)
  - unsigned int HibernateIsActive (void)
  - unsigned long HibernateLowBatGet (void)
  - void HibernateLowBatSet (unsigned long ulLowBatFlags)
  - void HibernateRequest (void)
  - void HibernateRTCDisable (void)
  - void HibernateRTCEnable (void)
  - unsigned long HibernateRTCGet (void)
  - unsigned long HibernateRTCMatch0Get (void)
  - void HibernateRTCMatch0Set (unsigned long ulMatch)
  - unsigned long HibernateRTCMatch1Get (void)
  - void HibernateRTCMatch1Set (unsigned long ulMatch)
  - void HibernateRTCSet (unsigned long ulRTCValue)
  - unsigned long HibernateRTCTrimGet (void)
  - void HibernateRTCTrimSet (unsigned long ulTrim)
  - unsigned long HibernateWakeGet (void)
  - void HibernateWakeSet (unsigned long ulWakeFlags)

下载用户驱动程序库手册， www.ti.com/stellaris

Texas Instruments

更多信息？

- 文档
  - 器件的数据手册
    - www.ti.com/stellaris
  - 驱动程序库用户手册
  - 其他文档。
- 软件
  - 代码范例
  - 驱动程序库 API 等
    - www.ti.com/stellarisware
- 联系 Stellaris *专家!*
  - 培训与技术支持
    - e2e.ti.com
  - 市场营销信息
    - stellaris-marketing@ti.com

TI E2E 社区里的 Stellaris 内部论坛

Hands-on 实验题

请继续阅读第 2 节和第 3 节，了解动手实验课程方面的信息。

第 2 节：如何基于EK-LM4F232评估板开始休眠模块开发

第 3 节：练习

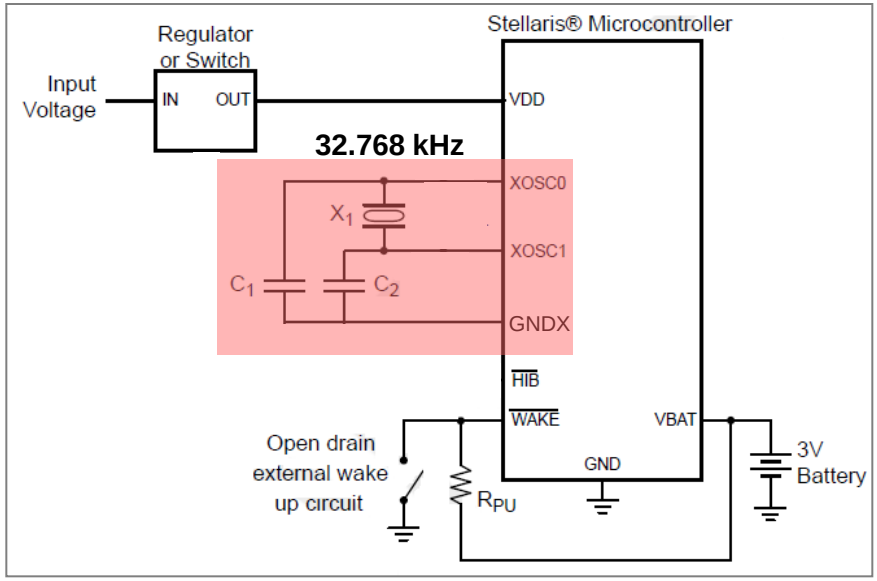
备用幻灯片

图像：www.12vpowerinverter.wordpress.com

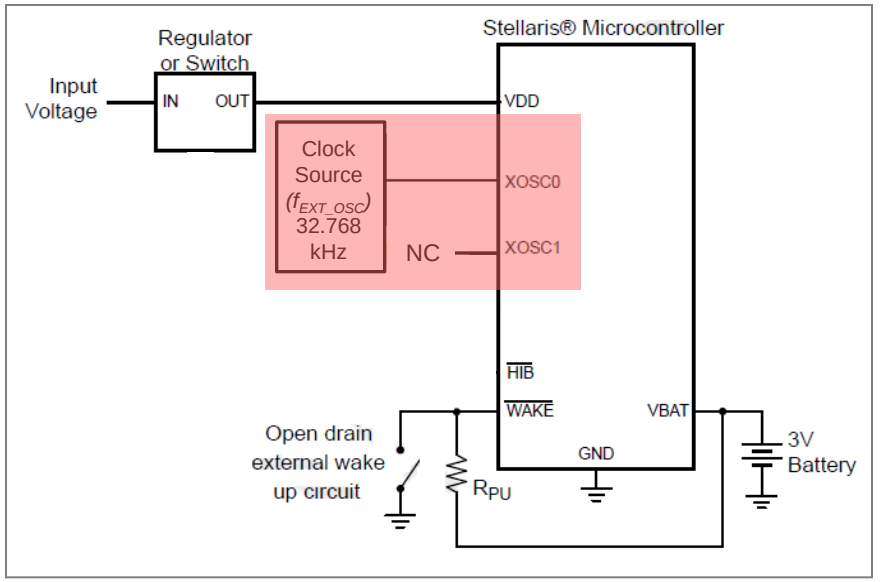
冬眠时钟源

备用

- 冬眠模块需要一个独立于主系统时钟的外部时钟源。
- 一个独立的时钟源需要在主系统时钟由于 V_{DD} 的移除而断电时保持 RTC/保存电池供电式存储器中的内容。
- 一个 32.768-kHz 晶体或一个外部时钟源可用于为冬眠模块提供时钟。
- 如果应用不需要冬眠模式，则 XOSC1 引脚可保持未连接状态，XOSC0 可连接至地，而 V_{BAT} 引脚应连接至V_{DD}。



冬眠时钟源：外部晶体



冬眠时钟源：外部振荡器

寄存器访问定时

备用

- 冬眠模块具有一种独立的定时机理，因此它在写访问之间需要一个定时间隙 (92µs, $t_{HIB\_REG\_ACCESS}$)
  - 在连续的写入操作之间
  - 在一个写入操作与随后的一个读取操作之间
  - 延迟必须由软件提供保护，可采用下面的任一种
  - HIBCTL 寄存器中的 WRC 位：
    - 如果 WRC [RO] = 0，则接口正在处理一个先前的写入操作，处于被占用状态
    - 如果 WRC [RO] = 1，则接口随时可以接受一个写入操作
  - HIBMIS 寄存器中的 WC 中断：
    - 如果 WC [RO] = 1，则由于正在被置位的 WRC 位的原因，发出的是一个未屏蔽的中断信号
    - 如果 WD [RO] = 0，则 WRC 位尚未被置位或中断被屏蔽
- 在背对背读取操作之间没有定时限制。读取操作以全系统时钟速率执行。

中断与唤醒

备用

- 当发生以下事件时，冬眠模块会生成中断：
  - WAKE 引脚被置为有效，
  - RTC 计数器与 *'match'* 值相匹配，
  - 在 V_{BAT} 引脚上检测到低电压（即：低电池电量状况），
  - 一个冬眠寄存器写入操作完成。
- 所有的中断均通过逻辑“或”组合在一起，因此只有一个中断请求可在某个给定的时刻传送至中断控制器。
- 可以检查 HIBRIS 寄存器以确定待处理事件的状态。
- HIBIM 寄存器可用于配置那些会触发中断的事件。
- 通过读取 HIBMIS 寄存器可服务于多个中断。