
在 STM32L0 和 STM32L4 系列微控制器中 使用 LPUART 使功耗最小

前言

STM32L0 和 STM32L4 系列微控制器具有复用的 UART 接口，这一设计令 MCU 仅需最小功耗即可工作。

本文的目的是解释怎样充分利用低功耗 UART（LPUART）的优势，扩展上述产品的电池寿命。

它显示了在实际示例中，等待通信设备的极低功耗。软件包 X-CUBE-LPUART 提供了在 [第 6.1 节](#)和 [第 6.2 节](#)中所述的用于执行测量的代码，可从 www.st.com 下载。

下列文件（也可在 www.st.com 获取）可作为参考：

- 应用笔记 AN4445：“STM32L0xx 超低功耗特性概述”
- 应用笔记 AN4621：“STM32L4xx 超低功耗特性概述”
- 应用笔记 AN4746：“优化 STM32L4xx 的功耗和性能”
- 参考手册 RM0351：“STM32L4x6 高级的基于 ARM® 的 32 位 MCU”
- 参考手册 RM0376：“超低功耗 STM32L0x2 高级的基于 ARM® 的 32 位 MCU”。

目录

1	定义	6
2	特性总结	7
2.1	L0 和 L4 系列比较	7
2.1.1	时钟子系统	7
2.1.2	电源管理	7
2.2	与 USART 外设的比较	7
3	工作模式	9
3.1	轮询模式	9
3.2	IT 模式	9
3.3	DMA 模式	9
3.4	组合模式举例	9
3.4.1	带轮询的中断	10
3.4.2	DMA 和直接访问组合	10
4	其他注意事项	11
4.1	从 SRAM 执行	11
4.2	GPIO 配置	11
4.3	时钟配置	11
4.3.1	时钟预分频器	12
4.4	功率配置	12
4.4.1	使用 Stop 和 Sleep 模式	12
4.4.2	运行时配置	13
5	可靠性和通信质量	14
5.1	噪声和频偏	14
5.2	丢字节	14
6	功耗比较	15
6.1	STM32L053 Nucleo 板测量	16
6.1.1	Stop vs. Sleep 模式	16
6.1.2	短时间 Sleep 模式和低功耗运行	17

6.1.3	中断操作开销	18
6.1.4	在接收字节之间转至 Stop	19
6.1.5	不同的振荡器时钟速度	21
6.1.6	更改 AHB 分频比	22
6.1.7	不同的外设时钟设置	23
6.1.8	ULP 位设置	25
6.1.9	更高通信速率	26
6.1.10	用 HSI, 从 Stop 模式唤醒	27
6.1.11	调压器设置	29
6.1.12	GPIO 上拉	30
6.2	STM32L476 Nucleo 板上的测量	31
6.2.1	三种方法一览	31
6.2.2	低内核频率时的简单轮询模式	32
6.2.3	调压器设置的作用	33
6.2.4	空闲模式比较	34
6.2.5	对于更高速率使用 MSI PLL 模式	36
6.2.6	使用两个振荡器	37
7	示例项目	39
7.1	硬件设置	39
7.2	配置示例	39
7.3	示例操作	39
8	结论	40
9	版本历史	41

表格索引

表 1.	缩略语列表	6
表 2.	特性比较	8
表 3.	时钟选项	12
表 4.	配置 - Stop vs. Sleep 模式	16
表 5.	配置 - Sleep 模式 vs. LPRUN	18
表 6.	配置 - 中断操作	19
表 7.	配置 - 在接收数据期间使用 Stop	20
表 8.	配置 - 内核时钟速度	21
表 9.	配置 - AHB 分频	23
表 10.	配置 - 时钟分频	24
表 11.	配置 - ULP 位的效果	26
表 12.	配置 - 更高通信速率	27
表 13.	配置 - HSI vs. MSI	28
表 14.	配置 - 调压器设置	29
表 15.	配置 - GPIO 上拉	30
表 16.	配置 - 管理通信	31
表 17.	配置 - 轮询	33
表 18.	配置 - 调压器设置	34
表 19.	配置 - 空闲模式	35
表 20.	配置 - HSI 和 PLL	36
表 21.	配置 - 对于 56700 波特速率使用 Stop	37
表 22.	文档版本历史	41
表 23.	中文文档版本历史	41

图片索引

图 1.	测试循环说明	15
图 2.	Stop vs. Sleep 模式	16
图 3.	Sleep 模式 vs. LPRUN	17
图 4.	中断操作开销	19
图 5.	在接收数据期间使用 Stop	20
图 6.	内核时钟速度比较	21
图 7.	降低振荡器速度与 AHB 分频对比	22
图 8.	三种不同的外设时钟分频设置	24
图 9.	ULP 位的效果	25
图 10.	工作于 57600 波特	26
图 11.	HSI vs. MSI 在 4 MHz	28
图 12.	调压器设置比较	29
图 13.	GPIO 内部上拉	30
图 14.	管理通信的三种不同方法比较	31
图 15.	最低设置时的轮询通信	32
图 16.	调压器设置	34
图 17.	空闲模式比较	35
图 18.	HSI 和 PLL 电流消耗	36
图 19.	对于 57600 波特速率使用 Stop	37

1 定义

表 1. 缩略语列表

术语	说明
LSE	低速外部时钟
LSI	低速内部时钟
HSE	高速外部时钟
HSI, HSI16	高速内部时钟
MSI	多速内部时钟源
UART	通用异步收发器
LPUART	低功耗 UART
MCU	微控制器
USART	通用同步和异步收发器
BT LE	低功耗蓝牙
CPU	中央处理单元（MCU 的一部分）
NVIC	嵌套向量中断控制器
DMA	直接存储器访问
TC	发送完成
RM	参考手册
SWD	串行线调试接口

2 特性总结

2.1 L0 和 L4 系列比较

虽然 STM32L4 和 STM32L0 系列上的 LPUART 外设实际上相同，但这两种产品有很大区别。除了 ARM® Cortex® 内核不同，还有其它区别影响了 LPUART 的效率。

2.1.1 时钟子系统

STM32L4xx MCU 可支持更高的时钟速度，这有效地降低了高要求任务的处理时间。MSI 内部时钟源也因此调整以达到更高时钟速度。这一 MSI 的加速降低了低速的选择范围，STM32L4 MSI 的最低可能频率为 100 kHz，而 STM32L0xx MCU 为 65 kHz。

1 MHz 以下的可选速度是不同的，因此难以直接比较两种 MCU 的效率。另外，在 STM32L4 系列中，MSI 可在其 PLL 模式中使用 LSE 硬件自动校准。这使得 MSI 更加精确，而 STM32L0 产品无此选项。

STM32L0 系列对于 HSI 时钟源具有一个简单的时钟因子 4 分频器，这使得 HSI 成为 16 或 4 MHz 的有效源。STM32L4 MCU 在 HSI16 时钟上没有直接的分频器。因此，在一些 UART 速度需要高于 9600 Bd 的应用中，STM32L0 可能会高效很多。对于 STM32L4，若 16 MHz 不够有效，解决方案为使用第二时钟源做系统时钟。

LSI 时钟频率也有不同，但这与我们的情况无关。

2.1.2 电源管理

STM32L4 系列的主调压器仅有两个范围，而 STM32L0xx MCU 有三个。此外，由于有更强的 MCU 支持，这两个范围被移到了更高频率。

非常有用的低功耗运行模式弥补了这一不足。

STM32L0xx MCU 在从 Sleep 或 Stop 低功耗模式唤醒后，无法直接返回至低功耗运行模式，而 STM32L4 系列不存在这一限制。STM32L4xx 上的低功耗运行也不限于 MSI 范围 1，还能工作于高达 2 MHz 的系统时钟速度。即使在低功耗运行中，HSI16 仍可被用作外设时钟。

2.2 与 USART 外设的比较

LPUART 比 USART 特性更少，但可用更低功耗工作，并可更有效地使用 LSE 时钟。

两种外设的主要特性总结于 [表 2](#) 中。

表 2. 特性比较

配置	LPUART	USART
LSE 9600 波特选项	+	-
同步模式	-	+
Ir SIR 兼容性	-	+
智能卡模式	-	+
自动波特率检测	-	+
Modbus 通信	-	+
LIN 模式	-	+

USART 还可使用 LSE 时钟工作，当过采样为 8 时，通信速度最高为 4000 波特，当过采样为 16 时，为 2000 波特。

当使用 LPUART 时，仅需 32.768 kHz LSE 时钟即可达到 9600 波特的串行通信，功耗最低，外部晶振可允许非常精确的速度设置。

对于更高的速率（例如用于 BT LE 的 57600 波特），LPUART 能效优势下降，但仍可测量为 5% 左右（参见第 6.1.9 节：更高通信速率）。

本文侧重于 9600 波特处的通信，这对很多应用是通用的：当在 9600 波特速率设置处从 Stop 唤醒时，LPUART 的能效优势更加显著。



3 工作模式

实际场景可能覆盖多种配置，使用不同的波特率、发 / 收比以及消息间时延。所有这些因素都会影响工作模式的选择。

3.1 轮询模式

轮询模式为最简单的可能的工作模式，也称为阻塞模式。CPU 处理单个任务，在非激活期间切换至低功耗模式。几乎没有处理开销，因此可使用非常低的系统时钟速度。

对于非常简单的场景，此模式极其有效。然而阻塞模式有效地阻塞了 CPU，使其不处理任何其它任务，例如数据处理或并发通信。

3.2 IT 模式

第二个选项是完全依赖于中断，对于每个传输的字节都唤醒 CPU。此工作模式利用 ARM® Cortex® 高级 NVIC 的优势，将处理分割为原子操作，永不阻塞 CPU，实现实时响应。

然而，此模式会略微加重 CPU 负担，它增加了与栈和上下文恢复有关的处理开销。

3.3 DMA 模式

在 DMA 模式中，CPU 空余了一大部分的处理，设置 DMA 通道在外设和 SRAM 间移动数据。CPU 可能会花费一部分处理时间在 Sleep 模式。对于循环缓冲管理，用户应在不需要时禁用半缓冲中断，以让 CPU 内核休息更长时间。然而，DMA 无法与 Stop 模式组合使用，在进入 Stop 低功耗模式之前，应禁用所有 DMA 通道。

LPUART 传输的实现得益于批采集模式（BAM），在该模式中 MCU 处于 Sleep 或低功耗 Sleep 模式（CPU 时钟关闭）。通过将 Flash 存储器配置为掉电模式并关闭其时钟，以及仅对 DMA、LPUART、SRAM 提供时钟，可优化功耗。

3.4 组合模式举例

实际应用通常是以上所述模式的混合，开发者一直尝试在冲突的需求间达到最佳。下面的例子并不在附带的源代码中，只是为了抛砖引玉。

3.4.1 带轮询的中断

一些嵌入式系统可能不需要严格实时，在此情况下，可阻塞 CPU 一段时间以处理一个消息帧。尤其是对于发送的情况，通常用于处理所有 TC 中断的能量被保留，在阻塞模式中发送消息，这时时钟速度可降低，功耗调压器切换至低功耗模式（LPRUN）。如果 CPU 可在阻塞模式中处理消息，则大约仅需要一半时钟速度。之后恢复正常工作，最好在 Stop 模式中等待进入的应答中断。

3.4.2 DMA 和直接访问组合

DMA 通道方便发送数据，它在接收期间是节约功耗的。

它的缺点是，在 DMA 模式，LPUART 无法利用从 Stop 模式唤醒的功能。原因是，在唤醒事件之后，DMA 获取正在进行的通信有困难。对于所有可能长时间空闲的应用来说，这是一个严重的缺点。

然而，DMA 仍可用于发送，然后用阻塞或中断方法接收。在通信系统中，若进入的消息快速连续或时序可预期，基于 DMA 的接收也可能是一个有效的选项。

4 其他注意事项

若需低功耗优势的完整概述，请参考已经提及的 AN4445 和 AN4621。下列建议针对我们的情况和用例。

4.1 从 SRAM 执行

如果可从 SRAM 执行程序，则有一个选项可以关闭嵌入式 Flash 存储器（通过时钟门控），进一步降低功耗。

4.2 GPIO 配置

一些 GPIO 设置可能对功耗有极大影响，而其它设置则不然。

用于 UART 通信线的引脚应配置为它们的复用功能模式。如果最重要的目标是达到可能的最低功耗，则不建议激活上拉，然而，在一些应用中，可能有必要增强通信的可靠性。速率设置与被测波特率处的功耗没有因果关系。

在半双工模式适用的地方，也可能做进一步的功耗节省，然而本文不涉及此配置。

其它应用不用的引脚应配置为模拟输入。一旦应用可以部署后，开发者不要忘记将调试线置为模拟。

4.3 时钟配置

对于 LPUART 外设时钟，为达到 9600 波特并从 Stop 模式唤醒，应选择 LSE 源。

建议对于更高速率使用 HSI。

系统时钟显而易见的（也是默认的）选择通常是 MSI 振荡器。除此之外，STM32L0xx 微控制器提供了 HSI 时钟分支上简单的 4 因子分频器，以产生 4 MHz。在 STM32L4xx 产品上，MSI 提供了更高的灵活性，可在 LSE 为 PLL 模式时，使用硬件自动校准纠正它的波动。

可使用预分频器和 PLL 生成其它速度，本文随后会分析不同的选项。当功耗为第一考虑因素时，这些解决方案都不是最佳的。用户必须参考产品数据手册，以得到不同时钟配置的典型功耗。

表 3. 时钟选项

源	优点	限制
HSI/HSI16	可使用具有精确调节的高速时钟	具有最差 MHz / W 比值的选项 在 L4 系列为固定速率
MSI	简单快速的时钟限速 最低的总体功耗（对于 L0，时钟低于 1 MHz 时）	时钟相对不稳定、不准确（除非在 LSE 为 PLL 模式时使用自动校准）
HSE	对于 L0，可能在 1 MHz 及以上有最高的功耗效率 其效率与 L4 上的 MSI 可以比拟	需要额外的外部组件 速率控制选项有限（仅预分频器）

4.3.1 时钟预分频器

RCC 模块提供了关闭外设总线时钟、AHB 和 APB 频率的可能性（RCC_CFGR 寄存器）。

降低 AHB 可带来极大功率节省，但在多数情况下不值得使用。存储器接口也由 AHB 提供时钟。虽然预分频为 8 时功耗可降低约 50%，但因为从程序存储器获取指令并将数据存储至存储器的时延，所以系统处理能力降低更多。通常，最好是降低系统时钟，而不是设置 AHB 预分频器（第 6.1.6 节）。

降低 APB 会限制 LPUART 外设和内核之间的带宽。因为传输的数据量低，所以这通常不是问题。尤其是对于 DMA 传输的情况，这不是问题。对于 CPU 驱动传输的情况，内核可能停止，等待总线传输完成。在极限情况下，这会导致工作的总能耗预算增加。此外，若总线传输未完成，则会防止内核进入低功耗模式，如 Stop 或 Sleep 模式。对于本应用笔记自带的 LPUART 示例，通常设置 APB 预分频器值为 4 是安全的，但更高的值可能导致问题，且不会带来功耗优势（第 6.1.7 节）。

4.4 功率配置

本节涉及的是当为 LPUART 通信配置功率模式时的实践意义，若需完整信息，请阅读 RM0376 和 RM0351 的专门章节。

4.4.1 使用 Stop 和 Sleep 模式

嵌入式软件的一个范例是，在空闲期间，CPU 不应该运行及主动检查标志，而是应该切换至低功耗模式，挂起时钟，只有外部中断事件时才恢复工作。

对于 ARM® MCU，可通过执行 WFE 或 WFI 指令实现。STM32 微控制器的 L0 和 L4 系列提供了高度可配置的空闲模式选择，在 RM 中详细说明。LPUART 的极大优势在于它能在等待接收消息时利用 Stop 模式（第 6.1.1 节）。

若运行在 STM32L0xx 产品上的应用不经常使用 VREFINT，且多数时间花费在低功耗模式，则请配置带快速唤醒的超低功耗模式（PWR 配置寄存器中的 ULP 和 FWU 位）。即使在使用一些内部电压参考的应用中，将其关闭，仅当需要测量时才检查启动时间也可能是很有意义的。然而请注意，VREFINT 的启动需要一些额外能耗。对于短时间的低功耗模式（即，典型的字节间的 Sleep 模式），将其关闭再开启没有意义，因为总的能耗预算会增长。

注： 不带快速唤醒的超低功耗模式无法保持 LPUART Tx 寄存器馈送，也无法赶得上阻塞和中断模式中进入的数据，即使是在 9600 波特的通信速率时。

STM32L4 系列没有关闭 VREFINT 的选项。

在多数情况下，最好在处理单个字节间激活 Sleep 模式。对于接收数据的情况，甚至可用 Stop 模式（第 6.1.4 节）。此实践在高时钟速度时可提供极大的优势，但不推荐在通信所需的最小内核时钟速度处使用。将设备置于 Sleep 模式并唤醒的开销会令功耗略微增加，不会被低功耗模式花费的非常短的时间平衡（第 6.1.2 节）。

4.4.2 运行时配置

调压器设置非常重要。请尽可能使用内核电压调节（参见 RM0351 和 RM0376 中专用的图）。在 STM32L0xx 上，从范围 1 切换到范围 3 可以简单地将典型功耗降低 25% 以上（第 6.1.11 节）。

STM32L4xx 仅提供了两个主调压器范围，但 STM32L0 和 STM32L4 系列都提供了低功耗运行模式。

STM32L0xx MCU 的轮询模式需要最小的 CPU 处理资源，可以利用低功耗运行模式。此模式本质上旁路了主调压器，其内核功耗能在主功耗调压器范围 3 的基础上再降低约 25%（第 6.1.2 节）。

在这些产品上，从 Stop 或 Sleep 模式唤醒会将主调压器恢复为开，考虑到当前的低时钟速度（最多是 MSI 范围 1），内核必须立即将所有处理能力分配给数据接收。实际上，LPRUN 对于发送阶段非常好，但当等待数据接收时，通常最好是利用 Stop 模式，而不是将 CPU 保持在 LPRUN。然而，可以将主调压器临时置为范围 3（当已经切换到 LPRUN 之后，它必须为范围 2）。更改主调压器的配置比切换调压器更快，且更不容易导致进入消息丢字节 – 从 Stop 唤醒及功率配置变化必须在 1 字节时间之内处理。

在 STM32L4xx 产品上，LPRUN 模式提供了更多的功能。它可工作于高达 2 MHz，这为中断模式 57600 波特通信速率提供了足够的处理能力。甚至可以从 Stop 唤醒，或直接 Sleep 至低功耗运行，而不唤醒主调压器。

5 可靠性和通信质量

5.1 噪声和频偏

对于典型低功耗应用的低通信速率，线性噪声应该不会导致问题。在一些情况下，可增加一个弱上拉电阻来减轻随机噪声。当然这会增加耗电。例如，在 [第 7 节](#) 中示例的发送期间，内部上拉（无额外元件）会增加约 30 μA 功耗。

通信参与者之一的基频（速率设置）不当会产生帧错误。最适合低功耗应用的 LSE 时钟源非常精确。仅当 LPUART 外设时钟由 MSI 时钟生成时，才很可能有帧错误。LPUART 也不能检测其对端未校准的通信速率，而是仅依靠精确的速率设置达到帧的可靠性。

5.2 丢字节

实现低功耗的主要方法是调低内核频率，使用低功耗模式。这对唤醒并响应外设事件提出了挑战。

对于发送情况，这会导致发送字节间的时延，加长激活时间，影响系统效率，但不会有数据损失。

真正的问题在于数据接收。在轮询或中断驱动模式，若没有在新数据接收之前及时读取 LPUART 数据寄存器中的接收数据，则会发生过载错误，在过载期间接收的任何数据都会丢失。

为避免接收数据丢失，提供了下述指导：

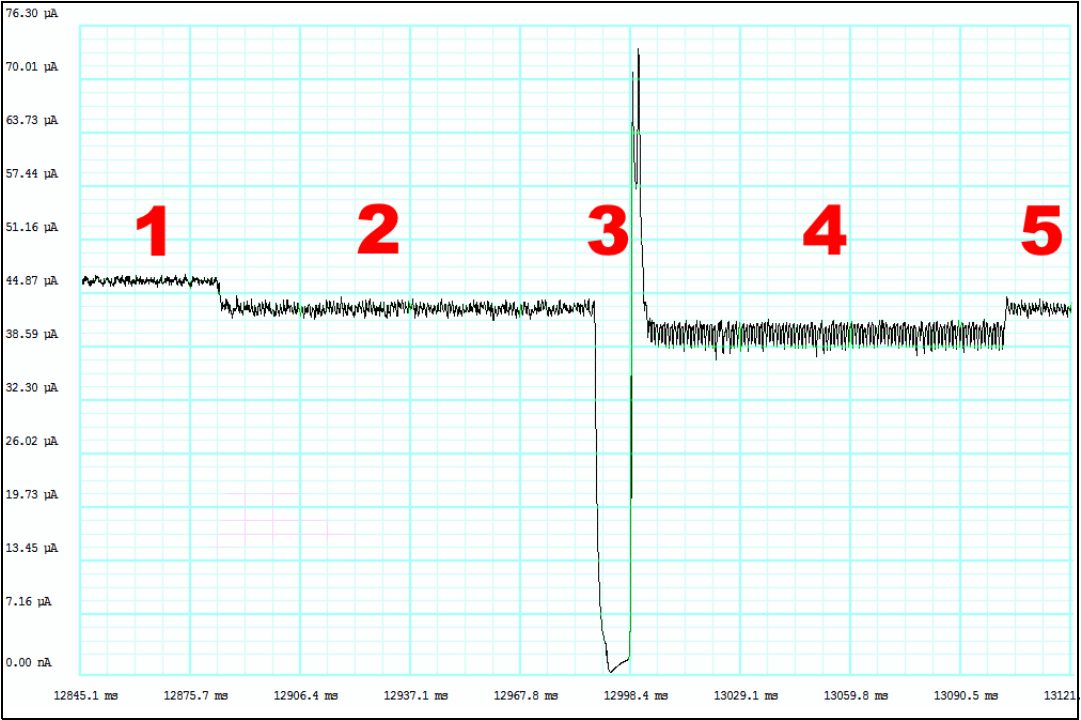
- 确保内核时钟速度正确（不要低于某个最低值）；
- 将 LPUART 中断的优先级调为应用中可能的最高级别，令其基本上不会被中断；
- 使用 DMA 做数据传输。

6 功耗比较

本节中的所有图片显示的是具体测量到的不同配置
低功耗 Nucleo 运行如下程序，图示于图 1 中：

1. 等待循环中的运行模式 – 显示常规功耗。
2. 发送 100 字节消息。
3. 空闲状态 10 ms: Stop 或 Sleep 模式等待应答（取决于配置）。
4. 接收相同的 100 字节。
5. 比较消息，检测通信错误，返回至 1。

图 1. 测试循环说明



设备开始时全速运行循环，显示其时钟配置的常规功耗。接下来是约 100 ms 的 9600 波特发送，然后是 10 ms 空闲状态，等待进入的应答。仅少数测量可具有 57600 的高通信速率，其中通信阶段 2 和 4 明显更短。接收应答需要另外的 100 ms。

发送和接收通常都比常规的运行模式功耗更低，这是因为设备配置为在消息字节间的短时间节省功耗。

注：所有测量都在温度约 25°C、使用 Nucleo 板执行，与数据手册的典型值相符。然而，不保证每一个 MCU 都能完全重现相同的数值。

6.1 STM32L053 Nucleo 板测量

6.1.1 Stop vs. Sleep 模式

第一个比较是 Stop 和 Sleep 模式间的区别。

设备尝试最小化等待进入消息期间的功耗。图 2 中的黑线表示设备在 Sleep 模式中等待的调试配置。使用 LSE 时钟和 LPUART 外设，Stop 模式可用于最高 9600 波特速率，这就是绿线。最后是紫线，可以看到 Stop 模式中消耗的绝对最低电流，其中调试线配置为模拟输入，设备使用 ULP + FWU 组合。

图 2. Stop vs. Sleep 模式

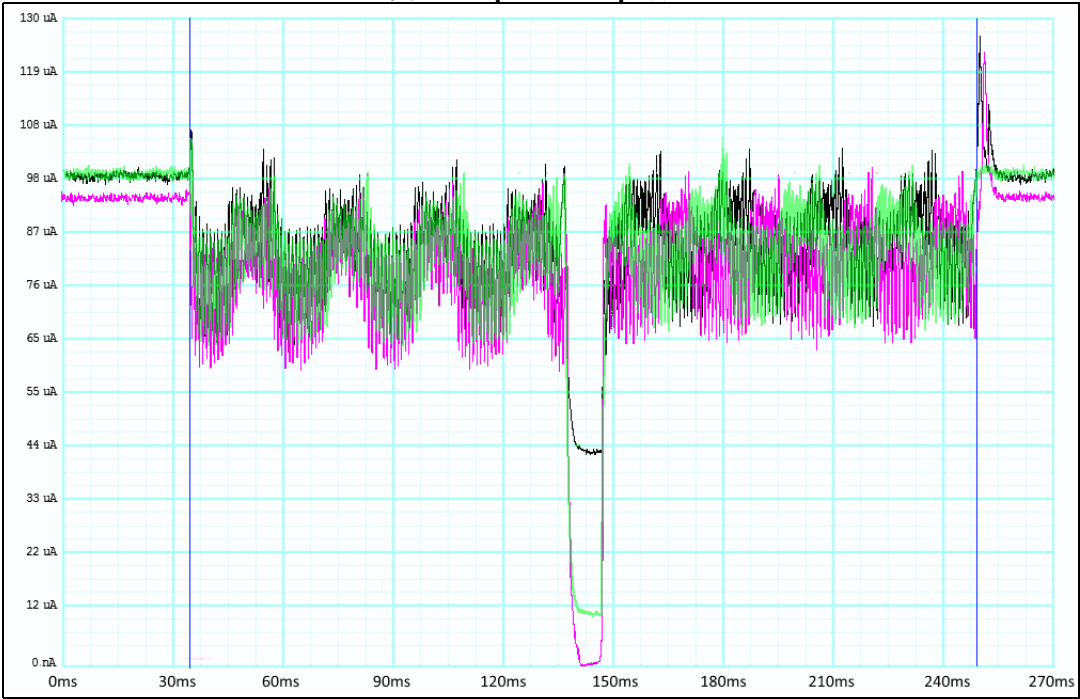


表 4. 配置 - Stop vs. Sleep 模式

图 2 中的曲线	黑色	绿色	紫色
发送间的空闲模式	Sleep	Stop	
调试接口	开启		关闭
平均电流 2、3、4 阶段	83.3 μ A	80.3 μ A	77.3 μ A
时钟	500 kHz MSI		
调压器	Range 3		
ULP/FWU	+/+		
AHB/APB 比	1/1		
波特率	9600		
使用的 SW	中断操作 (Com IT)，定义了 RXSLEEP 和 TXSLEEP		

6.1.2 短时间 Sleep 模式和低功耗运行

在大多数情况下，当等待标志或事件时，值得使用 WFE 将内核置于挂起模式，在一些情况下它也可能对优化功耗不利。当内核时钟速度刚刚够执行需要的操作时，增加 WFE 指令可能会实际上略微增加功耗。请考虑换为使用低功耗运行调压器模式。每次从 Stop 或 Sleep 模式唤醒都会终止低功耗运行的效果，但在唤醒之前它能节省很大功耗。

在图 3 中，红线表示发送每字节后转入 Sleep 模式的代码终止了低功耗运行模式的执行。绿线代码为调压器设为范围 3 运行，在发送期间不尝试使用 Sleep 模式。最后，黑线代码为在完成发送、转至 Stop 模式之前，停留在 LPRUN 的功耗。在接收阶段，所有三个代码都是在每字节之后，使用调压器范围 3 与 Sleep 模式。值得注意的是，对于这么短时间的 Stop 模式，将代码一直留在 LPRUN 会更有效。

图 3. Sleep 模式 vs. LPRUN



表 5. 配置 - Sleep 模式 vs. LPRUN

图 3 中的曲线	黑色	绿色	红色
发送间的空闲模式	Stop		
调试接口	关闭		
平均电流 2、3、4 阶段	34.9 μ A	40.0 μ A	42.0 μ A
时钟	130 kHz MSI		
调压器	LPRUN, 然后范围 3	Range 3	
AHB/APB 比	1/1		
ULP/FWU	+ / +		
波特率	9600		
使用的 SW	阻塞操作 (Com 轮询) 定义了 RXSLEEP		还定义了 TXSLEEP

6.1.3 中断操作开销

ARM® Cortex® 内核的中断机制非常有效，但仍然有额外的工作。通过比较适合我们示例代码中断驱动操作的最低内核时钟， 250 kHz，可最好地说明这一事实。

如图 4 中所示，IT 代码（黑线）实际上在 Sleep 模式（等待中断）几乎没有花费时间，功耗相对较高。红线表示的阻塞操作在等待事件（Tx 缓存空或 Rx 缓存满）上节省了一些功率。

因为执行的代码不同，此比较无法绝对公平，然而，它对代码优化非常有意义。



图 4. 中断操作开销

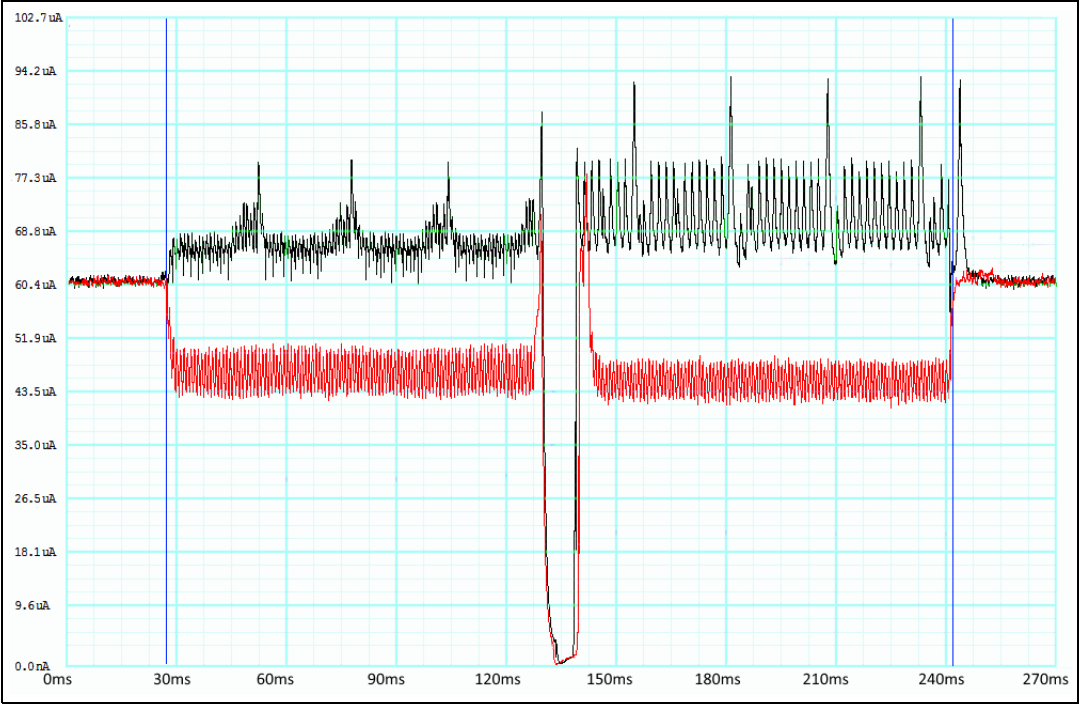


表 6. 配置 - 中断操作

图 4 中的曲线	黑色	红色
发送间的空闲模式	停止	
调试接口	关闭	
平均电流 2、3、4 阶段	66.6 μ A	44.3 μ A
时钟	250 kHz MSI	
调压器	Range 3	
AHB/APB 比	1/1	
ULP/FWU	+/+	
波特率	9600	
使用的 SW	中断操作 (Com IT) 定义了 RXSLEEP 和 TXSLEEP	阻塞操作 (Com 轮询), 定义了 RXSLEEP 和 TXSLEEP

6.1.4 在接收字节之间转至 Stop

LPUART 可在有外部事件时从 Stop 唤醒，这令它即使在接收阶段的字节之间停止时，也能进入 Stop 模式。

当工作速率稍低于调压器范围 3 的限值时具有最佳结果，具体来说就是 4 MHz MSI 时钟。在此方法中，中断开销时间非常短，MCU 多数时间花费在 Stop 模式。

速率低会导致在实际接收期间的功耗变得稍高。差别并不大，约 10%。然而，如果仍然使用轮询和很低的时钟设置，功耗可比这更低，但牺牲了为任何其它任务服务的灵活性。

图 5. 在接收数据期间使用 Stop

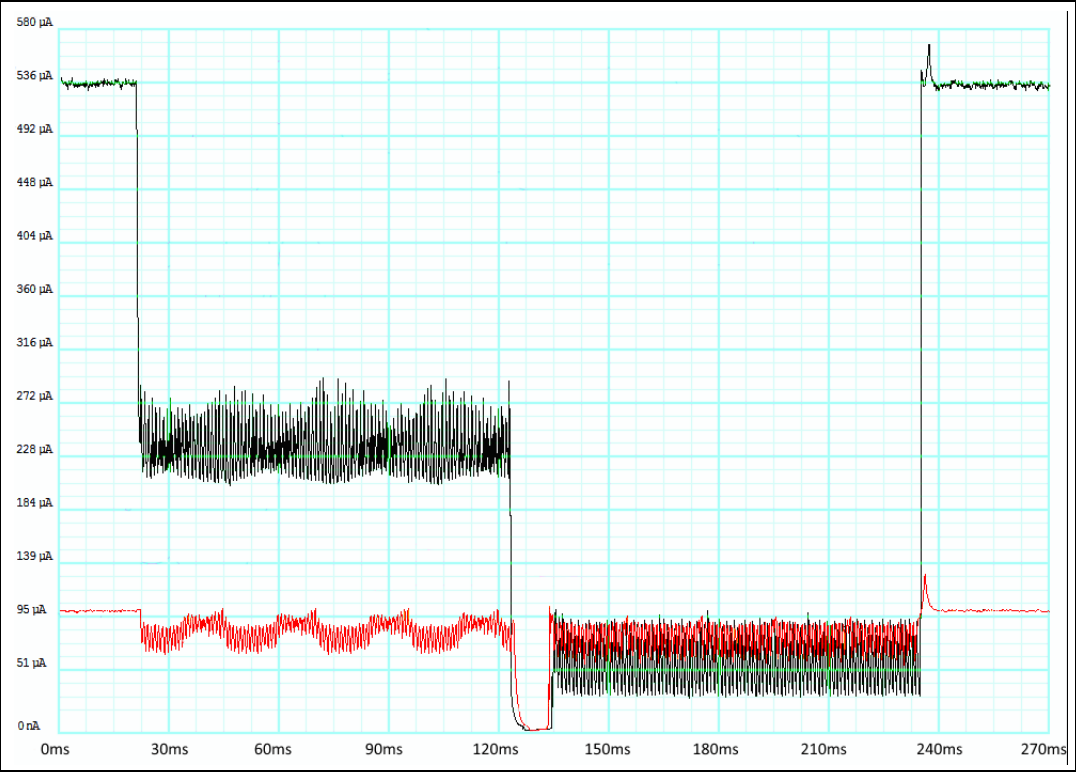


表 7. 配置 - 在接收数据期间使用 Stop

图 5 中的曲线	黑色	红色
发送间的空闲模式	Stop	
调试接口	关闭	
平均电流	55.5 µA （仅 Rx）	61.8 µA （仅 Rx）
时钟	4 MHz MSI	500 kHz MSI
调压器	Range 3	
AHB/APB 比	1/1	
ULP/FWU	+ / +	
波特率	9600	
使用的 SW	中断操作（Com IT），定义了 RXSTOP 和 TXSLEEP	

6.1.5 不同的振荡器时钟速度

使用 MSI 时钟，应用计算能力和功耗可轻易地调节到系统的当前需要。当然，可以在执行期间实时更改时钟设置。

[图 6](#) 显示了在 2 MHz 处执行的测试（黑线）、500 kHz 处（红线）、130 kHz 处（绿线）。差距很大，但在发送和接收字节间都使用 Stop 模式时，占空比使其不那么明显，速率越高，内核在低功耗状态花费的时间越长。

当然，在 Stop 模式期间没有明显差距。

图 6. 内核时钟速度比较

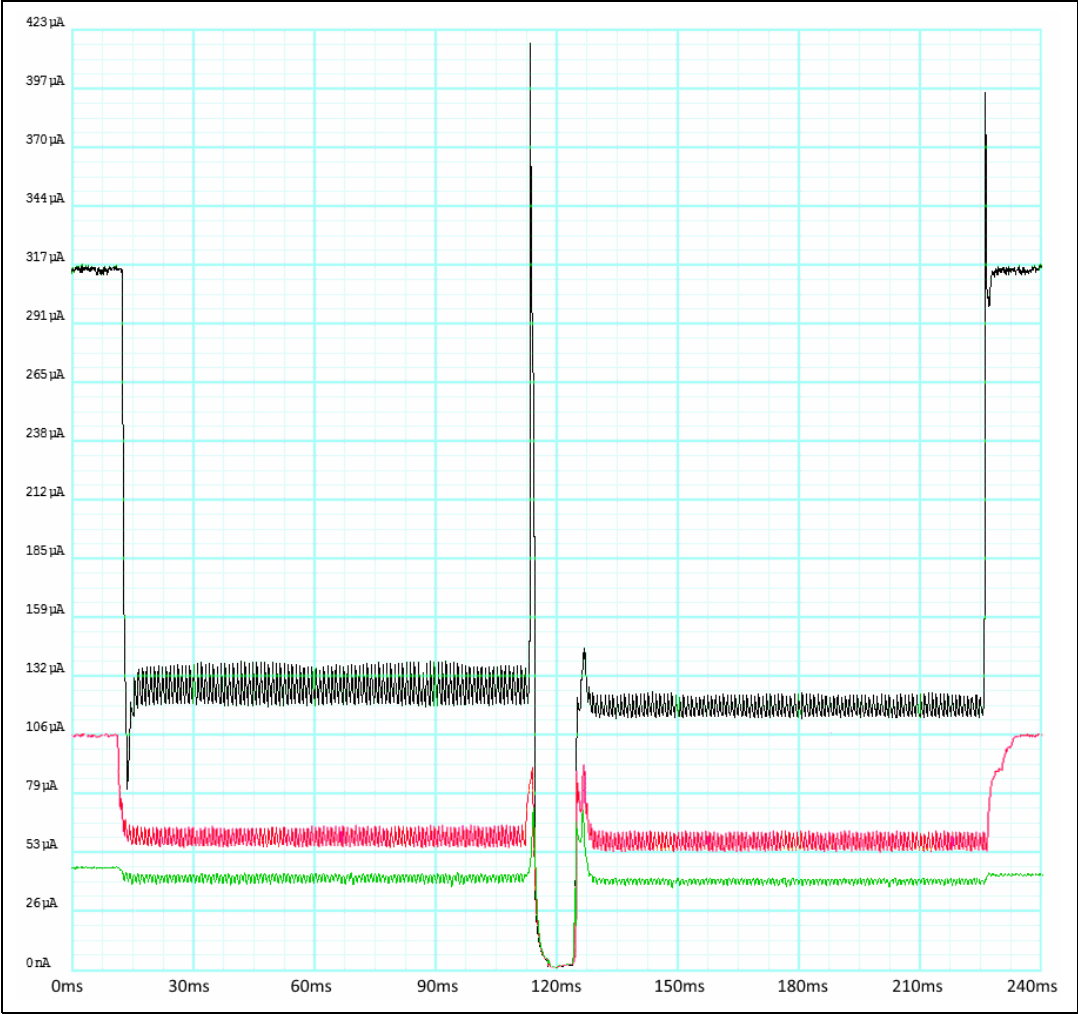


表 8. 配置 - 内核时钟速度

图 6 中的曲线	黑色	绿色	红色
发送间的空闲模式	Stop		
调试接口	关闭		
平均电流 2、3、4 阶段	136.6 µA	42 µA	57.1 µA
时钟	2 MHz MSI	130 kHz MSI	500 kHz MSI

表 8. 配置 - 内核时钟速度（续）

图 6 中的曲线	黑色	绿色	红色
调压器	Range 3		
AHB/APB 比	1/1		
ULP/FWU	+/+		
波特率	9600		
使用的 SW	阻塞操作（Com 轮询），定义了 RXSLEEP 和 TXSLEEP		

6.1.6 更改 AHB 分频比

更改 AHB 分频比会导致功耗降低，因为多数电路运行于低频。

这显示在图 7 中，其中黑线为示例运行于 1 MHz MSI 时钟时的电流消耗，红线为相同代码运行于 AHB 分频设为 4 时。若时钟可直接配置为低频，则选择此选项总是更好的。绿线表示相同示例，只是 MSI 振荡器直接设为 250 kHz。由 Stop 模式和运行阶段间的差距可证明，实际处理能力是相同的，但没有 AHB 分频的解决方案更有效。

仅在外部时钟情况下，或必须使用 HSI 时，使用 AHB 分频才有意义。

图 7. 降低振荡器速度与 AHB 分频对比

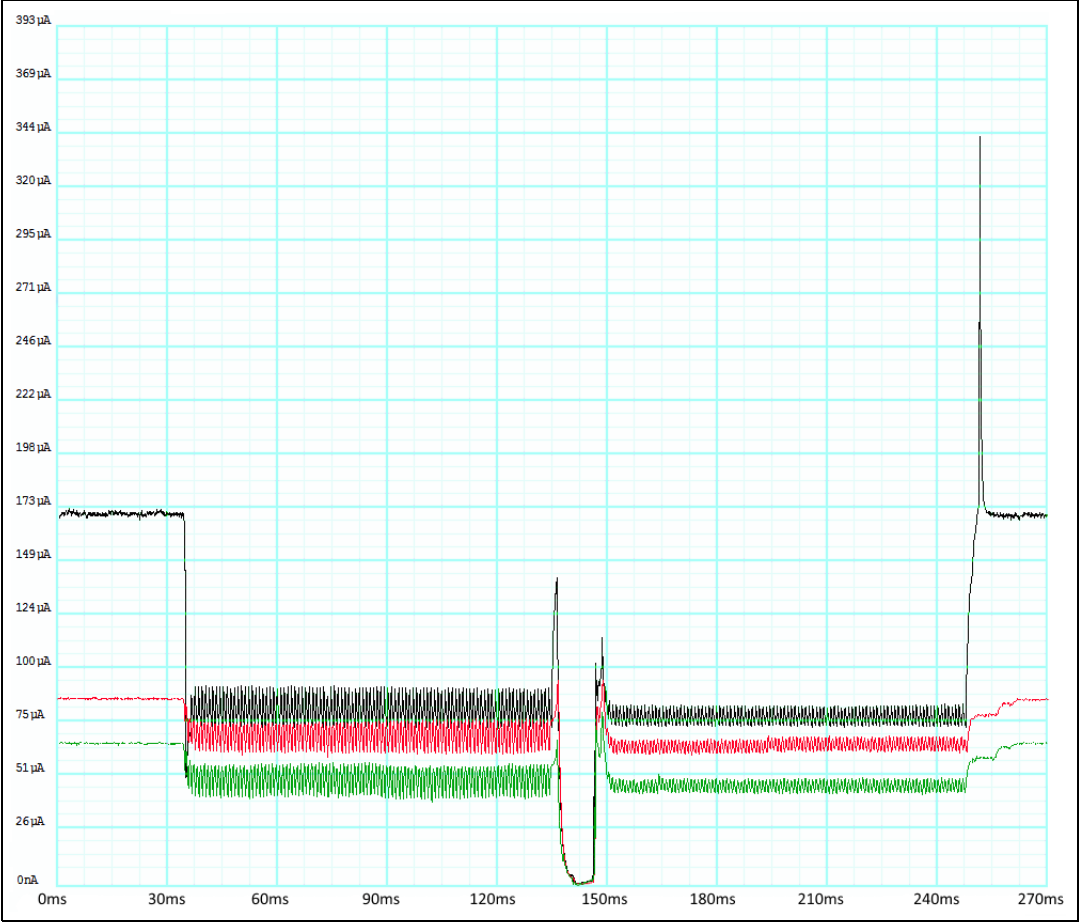


表 9. 配置 - AHB 分频

图 7 中的曲线	黑色	绿色	红色
发送间的空闲模式	Stop		
调试接口	关闭		
平均电流 2、3、4 阶段	75.9 μ A	44.3 μ A	63.9 μ A
时钟	1 MHz MSI	250 kHz MSI	1 MHz MSI
调压器	Range 3		
AHB/APB 比	1/1		4/1
ULP/FWU	+/+		
波特率	9600		
使用的 SW	阻塞操作（Com 轮询），定义了 RXSLEEP 和 TXSLEEP		

6.1.7 不同的外设时钟设置

在三种不同的 APB 时钟分频设置，测量基于 DMA 的通信，示于图 8 中，其中黑线表示无分频的电流消耗，红线和绿线分别是分频设为 4 和 8。

图 8. 三种不同的外设时钟分频设置

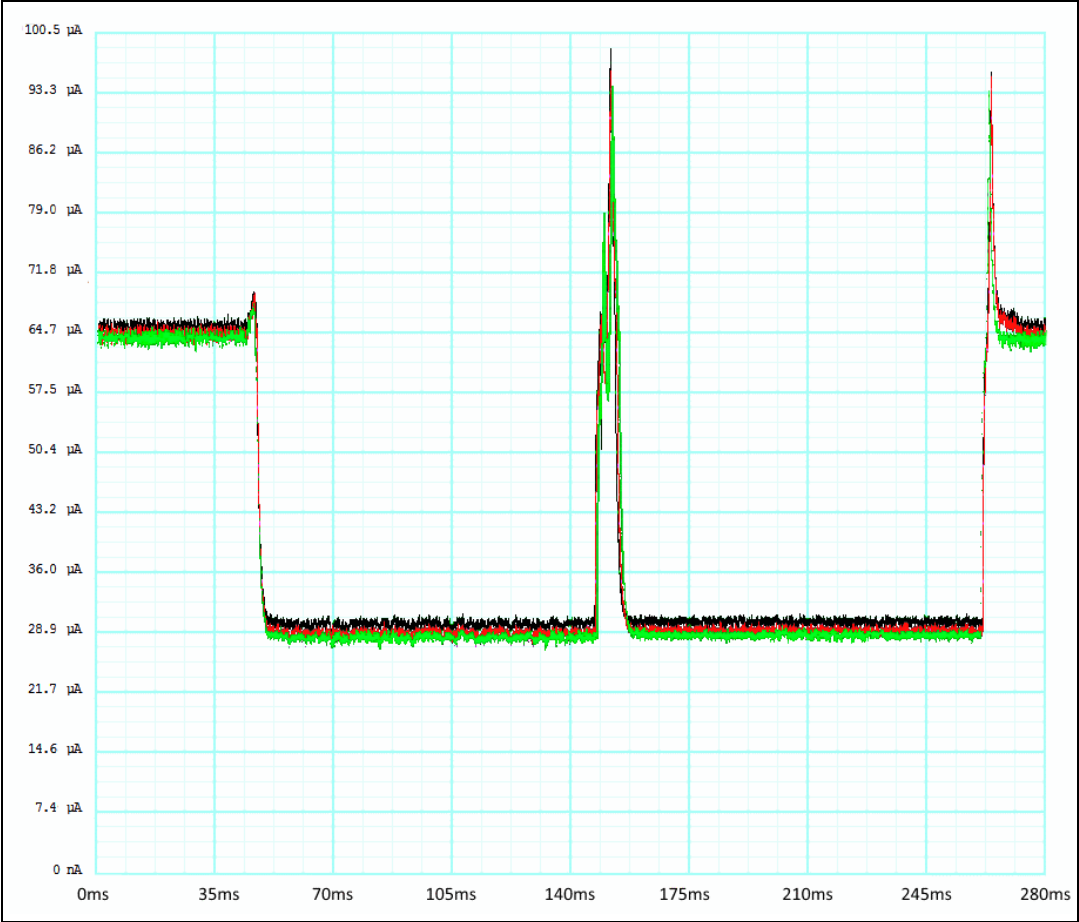


表 10. 配置 - 时钟分频

图 8 中的曲线	黑色	红色	绿色
发送间的空闲模式	Stop		
调试接口	关闭		
平均电流 2、3、4 阶段	31.5 µA	30.5 µA	30.1 µA
时钟	250 kHz MSI		
调压器	Range 3		
AHB/APB 比	1/1	1/4	1/8
ULP/FWU	+ / +		
波特率	9600		
使用的 SW	DMA 操作（Com DMA），禁用半缓冲中断		

DMA 工作示例的功率跟踪与中断或阻塞模式的代码执行不同。在发送的一半有一个峰值，它由半消息中断导致。对于应答数据接收的一半，内核停留于 Sleep 模式，它仅在后一半期间才激活。当然，不同的安排也是可能的。

6.1.8 ULP 位设置

ULP 位的角色是在 Stop 模式期间断开内部电压参考。[图 9](#) 中的功率分析截屏显示此设置（黑线）在 Stop 模式中达到了较低功耗，但当唤醒时产生了显著的峰值（与没有 ULP 的绿线相比）。

在 ULP 模式中等待进入数据的微控制器消耗约 $0.8\ \mu\text{A}$ ，相比起来没有 ULP 模式时几乎为 $3\ \mu\text{A}$ ，但如果频繁从空闲模式唤醒，由于电压参考的启动峰值，效果就不明显了。

图 9. ULP 位的效果



表 11. 配置 - ULP 位的效果

图 9 中的曲线	黑色	绿色
发送间的空闲模式	Stop	
调试接口	关闭	
平均电流 2、3、4 阶段	74.5 μ A	72.6 μ A
时钟	500 kHz MSI	
调压器	Range 3	
AHB/APB 比	1/1	
ULP/FWU	+/+	-/-
波特率	9600	
使用的 SW	中断操作 (Com IT)，定义了 RXSTOP 和 TXSLEEP	

6.1.9 更高通信速率

虽然在更高通信速率时 LPUART 没有更高级的节约功耗特性可用（与常规的 USART 外设相比），然而与全功能的 USART（黑线）相比，LPUART 简化电路仍有更低功耗（红线），参见图 10。

图 10. 工作于 57600 波特



表 12. 配置 - 更高通信速率

图 10 中的曲线	黑色	红色
发送间的空闲模式	Stop	
调试接口	开启	
平均电流 2、3、4 阶段	434.5 μ A	415.2 μ A
时钟	4 MHz MSI	
调压器	Range 1	
AHB/APB 比	1/1	
ULP/FWU	+/+	
波特率	57600	
使用的 SW	中断操作（Com IT），修改为 USART 在 SYSCLK	中断操作（Com IT），修改为 LPUART 在 SYSCLK

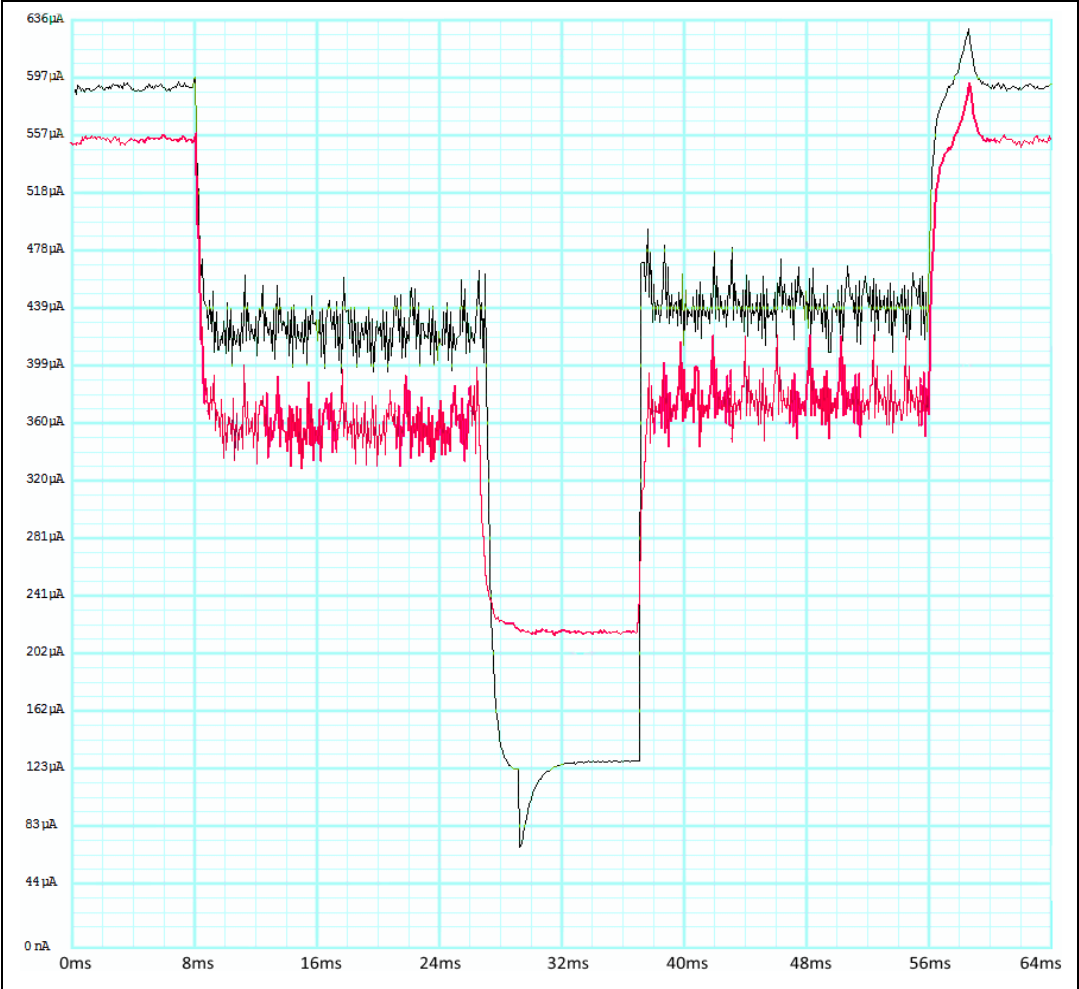
注：两种情况的时序略微不同。当用 MSI 作为 LPUART 外设的时钟时，这是一个常见问题。MSI 时钟一般不如 LSE 精确，甚至不如 HSI。

6.1.10 用 HSI，从 Stop 模式唤醒

使用更高通信速率的另一选项是利用从 Stop 模式唤醒的特性，由 HSI 时钟为外设提供时钟（参见图 11）。

HSI 振荡器与 Stop 模式组合消耗的电流大约和 MSI 在 Stop 模式中 2 MHz 处需要的相同。然而，它更加精确，为内核在运行模式提供了更大的处理能力。

图 11. HSI vs. MSI 在 4 MHz



请注意，在发送阶段，MSI 时序稍微有些偏离。当用 MSI 作为 LPUART 外设的时钟时，这是一个常见问题。

表 13. 配置 - HSI vs. MSI

图 11 中的曲线	黑色	红色
发送间的空闲模式	Stop	Sleep
调试接口	关闭	
平均电流 2、3、4 阶段	369.1 µA	329.7 µA
时钟	4 MHz HSI	4 MHz MSI
调压器	Range 3	
AHB/APB 比	1/1	
ULP/FWU	+ / +	
波特率	57600	
使用的 SW	中断操作 (Com IT)	



6.1.11 调压器设置

设置较低的内核电压是节能的一个简单直观方法。

在图 12 中证明，即使是 HSI 时钟，仅仅更改电压就能使运行模式显著“电流饥饿”。

请注意，该设置对空闲模式几乎或完全没有影响。

图 12. 调压器设置比较

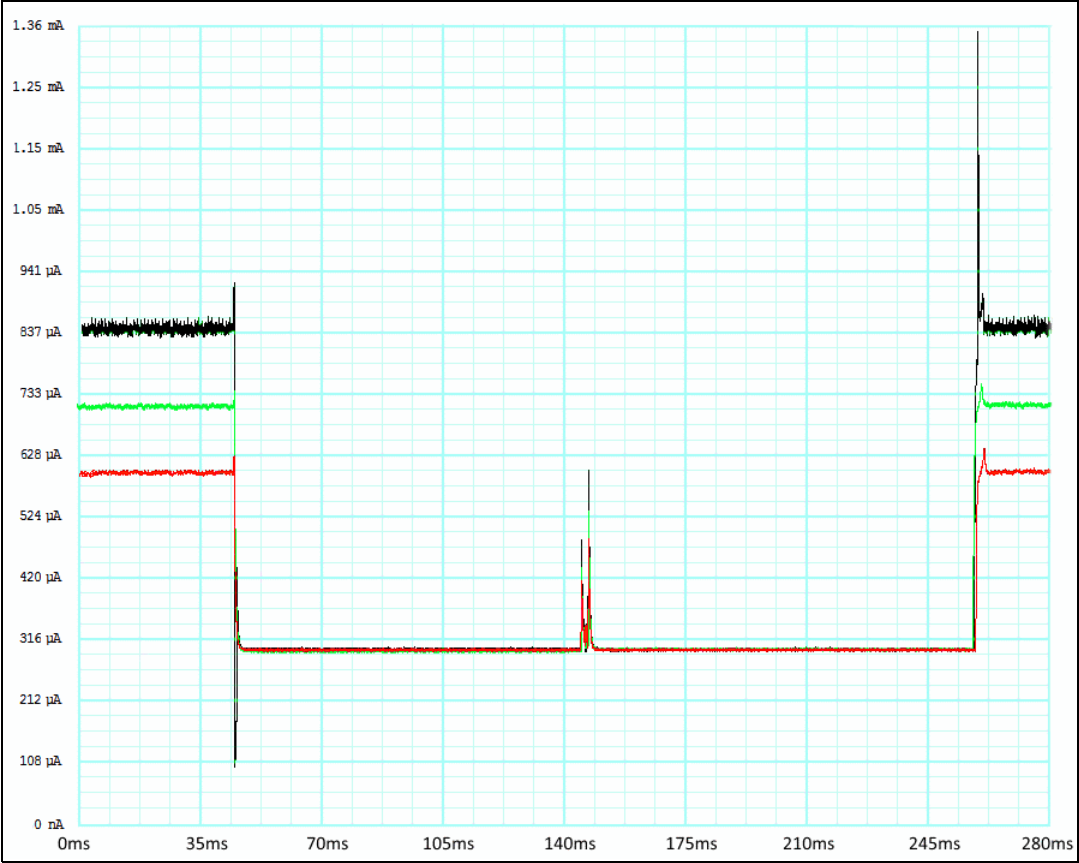


表 14. 配置 - 调压器设置

图 12 中的曲线	红色	绿色	黑色
发送间的空闲模式	Sleep		
调试接口	关闭		
平均电流 2、3、4 阶段	296.9 µA	300.8 µA	301.1 µA
时钟	4 MHz HSI		
调压器	Range 3	Range 2	Range 1
AHB/APB 比	1/1		
波特率	9600		
使用的 SW	DMA 操作 (Com DMA)		

6.1.12 GPIO 上拉

图 13 证明了启用 GPIO 上拉的能量成本，在此情况下是发送线。黑线是启用内部 GPIO 上拉，红线是没有上拉。
显而易见，没有其它阶段受到额外功耗的影响。

图 13. GPIO 内部上拉

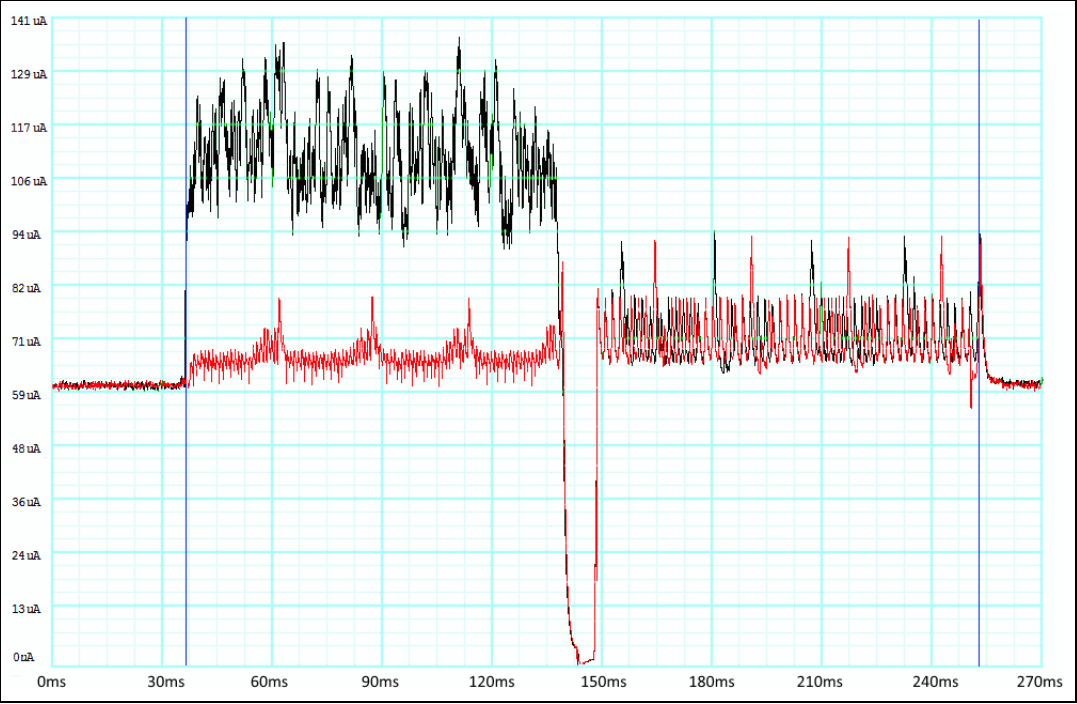


表 15. 配置 - GPIO 上拉

图 13 中的曲线	黑色	红色
发送间的空闲模式	Stop	
调试接口	关闭	
平均电流 2、3、4 阶段	87.8 μ A	66.6 μ A
时钟	250 kHz MSI	
调压器	Range 3	
AHB/APB 比	1/1	
ULP/FWU	+/+	
波特率	9600	
使用的 SW	中断操作 (Com IT)，修改为 Tx 线 内部上拉	中断操作 (Com IT)

6.2 STM32L476 Nucleo 板上的测量

6.2.1 三种方法一览

在本节中会比较管理通信的三种基本方法（DMA、轮询和中断）。

观察图 14，DMA 方法在比较中胜出，但是如果应答前的停止很长，它则会因为余量过宽而输掉。那么，轮询将是最有效的方法，但它假设只处理单个任务。中断看起来能解决这一问题，但对于 200 kHz 系统时钟，它没有余量做到这点（通过字节间 Sleep 时间可以忽略判断）。

虽然 DMA 和轮询驱动通信仍然工作于 100 kHz 的最小 MSI 设置，中断操作则不然。

图 14. 管理通信的三种不同方法比较

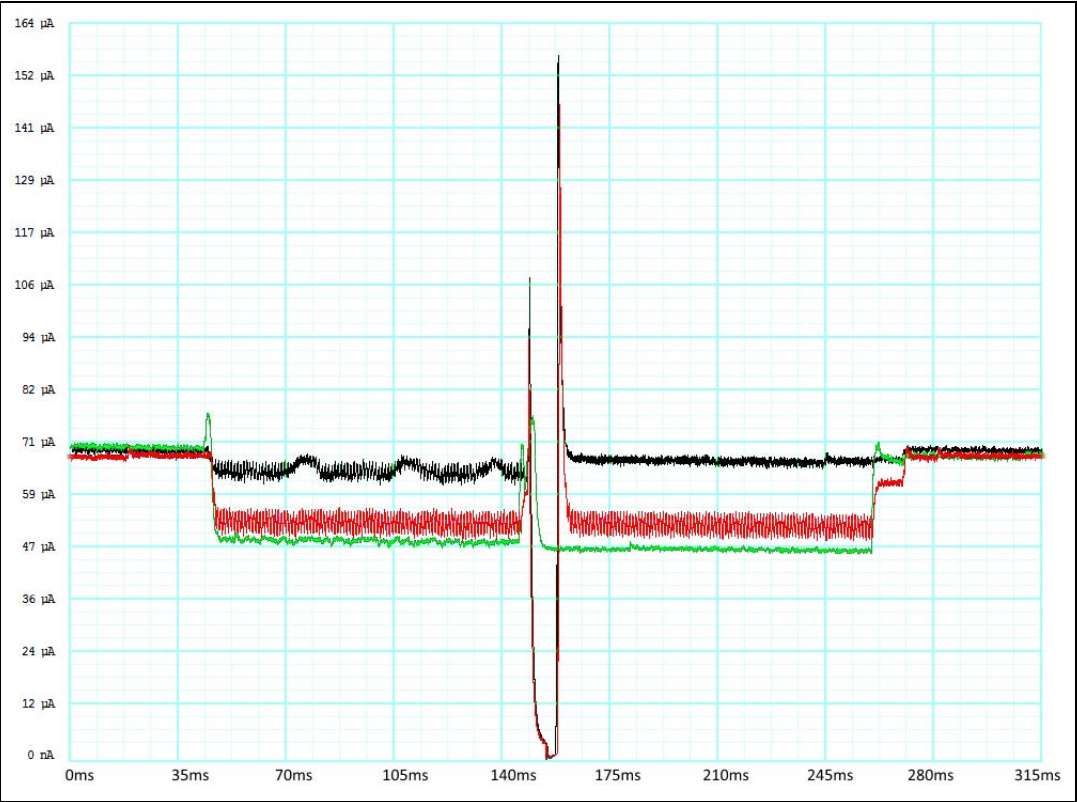


表 16. 配置 - 管理通信

图 14 中的曲线	黑色	红色	绿色
空闲模式	Stop 2		Sleep
调试接口	关闭		
平均电流 2、3、4 阶段	64.1 µA	51.6 µA	49.4 µA
时钟	200 MHz MSI		
调压器	低功耗运行		
AHB/APB 比	1/1		

表 16. 配置 - 管理通信 (续)

图 14 中的曲线	黑色	红色	绿色
波特率	9600		
使用的 SW	中断驱动 (Com IT)，在字节间 Sleep	轮询操作，定义了 TXSLEEP 和 RXSLEEP	DMA 数据流 (Com DMA)，有半缓冲中断

6.2.2 低内核频率时的简单轮询模式

复位之后，MCU 使用运行于 4 MHz 的 MSI 振荡器初始化，这是低功耗和可接受通信速率间的折中。需要低频率来使用 LPUART 建立通信链路。请注意在这一步，从不同外设来的中断可能会打断正在进行的通信。

图 15 显示了使用各种低速率 MSI 振荡器的功耗曲线。

图 15. 最低设置时的轮询通信

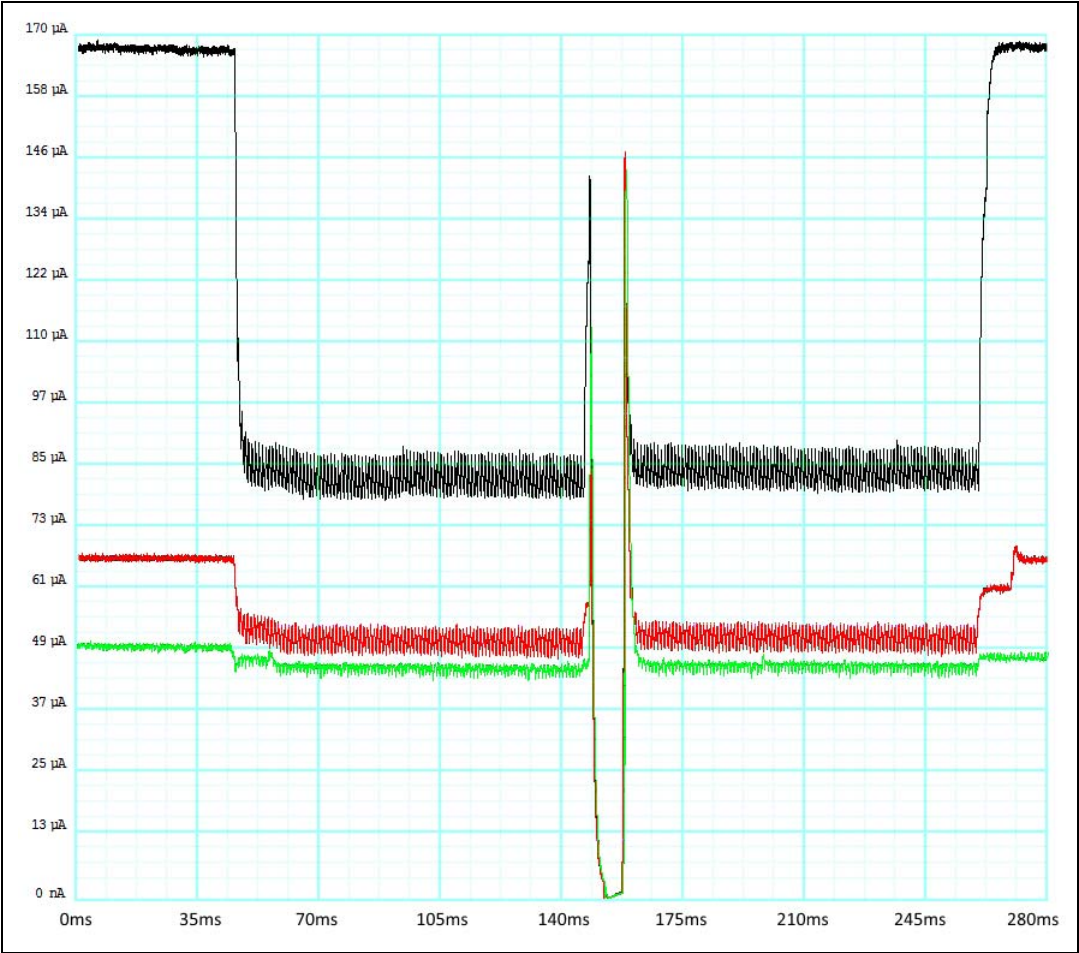


表 17. 配置 - 轮询

图 15 中的曲线	黑色	红色	绿色
空闲模式	Stop 2		
调试接口	关闭		
平均电流 2、3、4 阶段	81.2 μ A	49.7 μ A	44.9 μ A
时钟	800 kHz MSI	200 kHz MSI	100 kHz MSI
调压器	低功耗运行		
AHB/APB 比	1/1		
波特率	9600		
使用的 SW	轮询操作，定义了 TXSLEEP 和 RXSLEEP		

6.2.3 调压器设置的作用

通往低功耗的重要一步是使用适合实际内核频率的调压器设置。2 MHz 速率（MSI 范围 5）是低功耗运行模式所支持的最高设置。保持连接在主调压器（在图 16 中，甚至强调了在 Sleep 期间使用它）会浪费很大能量。

图 16. 调压器设置

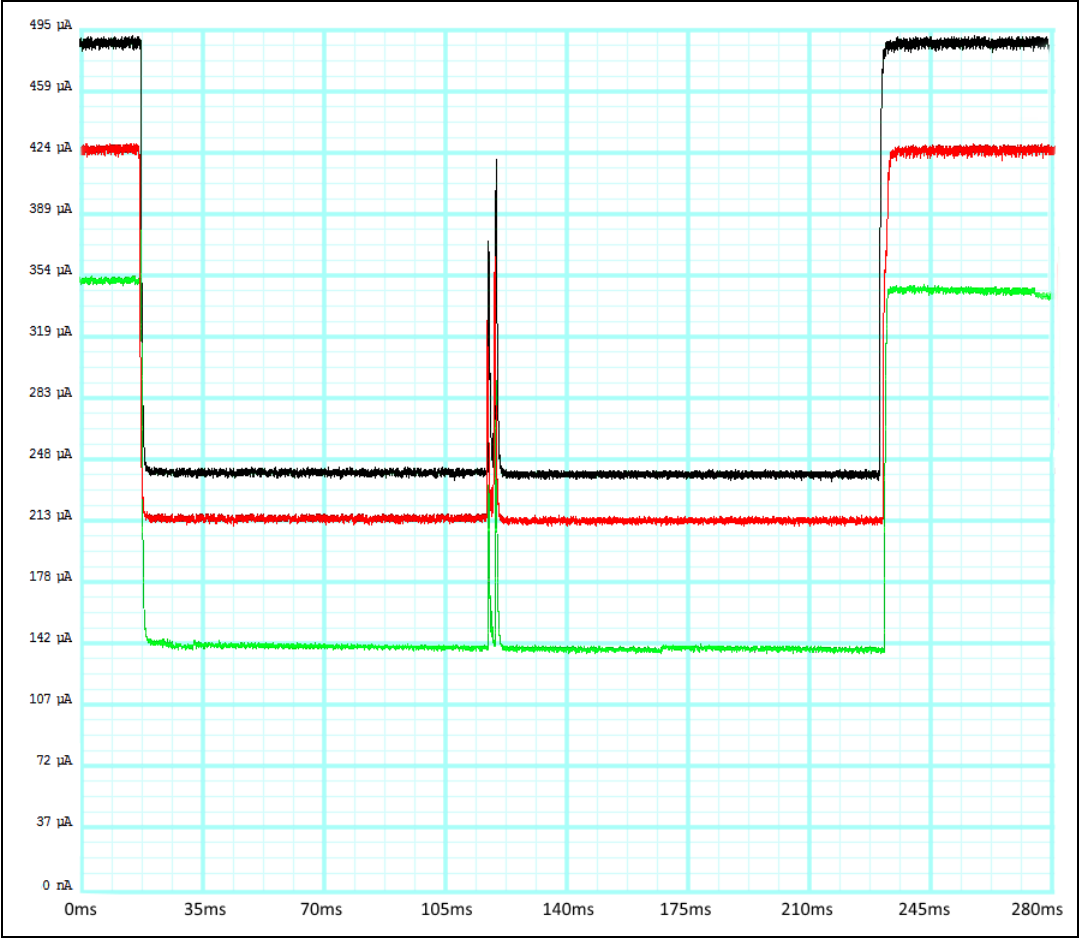


表 18. 配置 - 调压器设置

图 16 中的曲线	黑色	红色	绿色
空闲模式	Sleep		
调试接口	关闭		
平均电流 2、3、4 阶段	244.4 µA	215.5 µA	139.1 µA
时钟	2 MHz MSI		
调压器	Range 1	Range 2	低功耗运行
AHB/APB 比	1/1		
波特率	9600		
使用的 SW	DMA 操作（Com DMA），禁用半缓冲中断		

6.2.4 空闲模式比较

L4 系列微控制器具有三种基本的空闲模式，名为 Sleep、Stop 1 和 Stop 2。



请注意在图 17 中，Stop 2 如何极大地最小化了功耗，以及唤醒如何产生了显著的峰值。这会导致在约 10 ms 空闲的测量情况下，Stop 1 的表现超过 Stop 2。此峰值的主要原因是无法从低功耗模式直接进入 Stop 2 模式。当进入及离开 Stop 2 时，主调压器必须为开。

图 17. 空闲模式比较

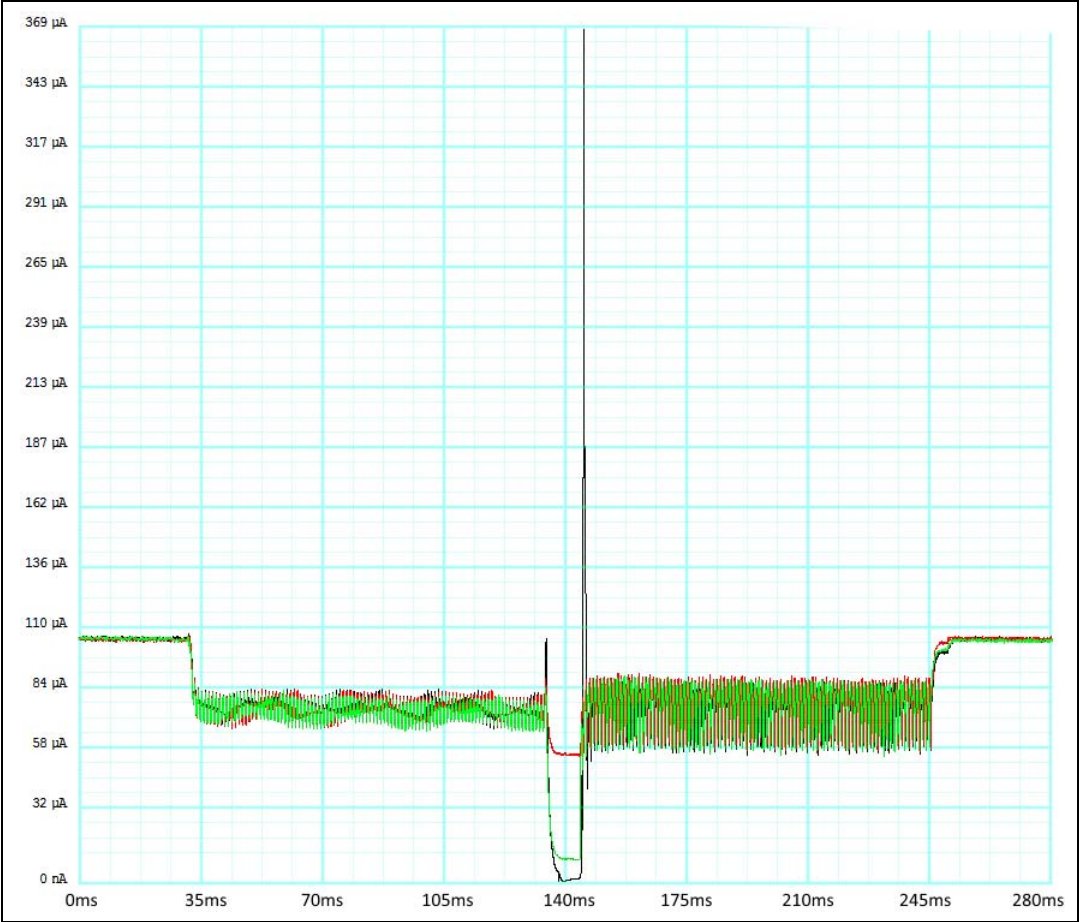


表 19. 配置 - 空闲模式

图 17 中的曲线	黑色	红色	绿色
空闲模式	Stop 2	Sleep	Stop 1
调试接口	关闭		
平均电流 2、3、4 阶段	73.2 µA	75.6 µA	72.8 µA
时钟	400 kHz MSI		
调压器	低功耗运行		
AHB/APB 比	1/1		
波特率	9600		
使用的 SW	IT 操作 (Com IT)，Rx Stop		

6.2.5 对于更高速率使用 MSI PLL 模式

要使用更高通信速率，需要有与 LSE 不同的 LPUART 时钟源。

可使用 MSI 驱动的最高 48 MHz 的系统时钟，配置为 PLL 模式，使用 LSE 自动校准。一个明显但效率低的解决方案是使用带分频器的 HSI 时钟（例如 AHB）。

HSE 外部源是另一选项。显示于图 18 中的测量比较了 HSI 配置与为得到低时钟频率及相应的低功耗而使用的 MSI PLL 模式。

图 18. HSI 和 PLL 电流消耗

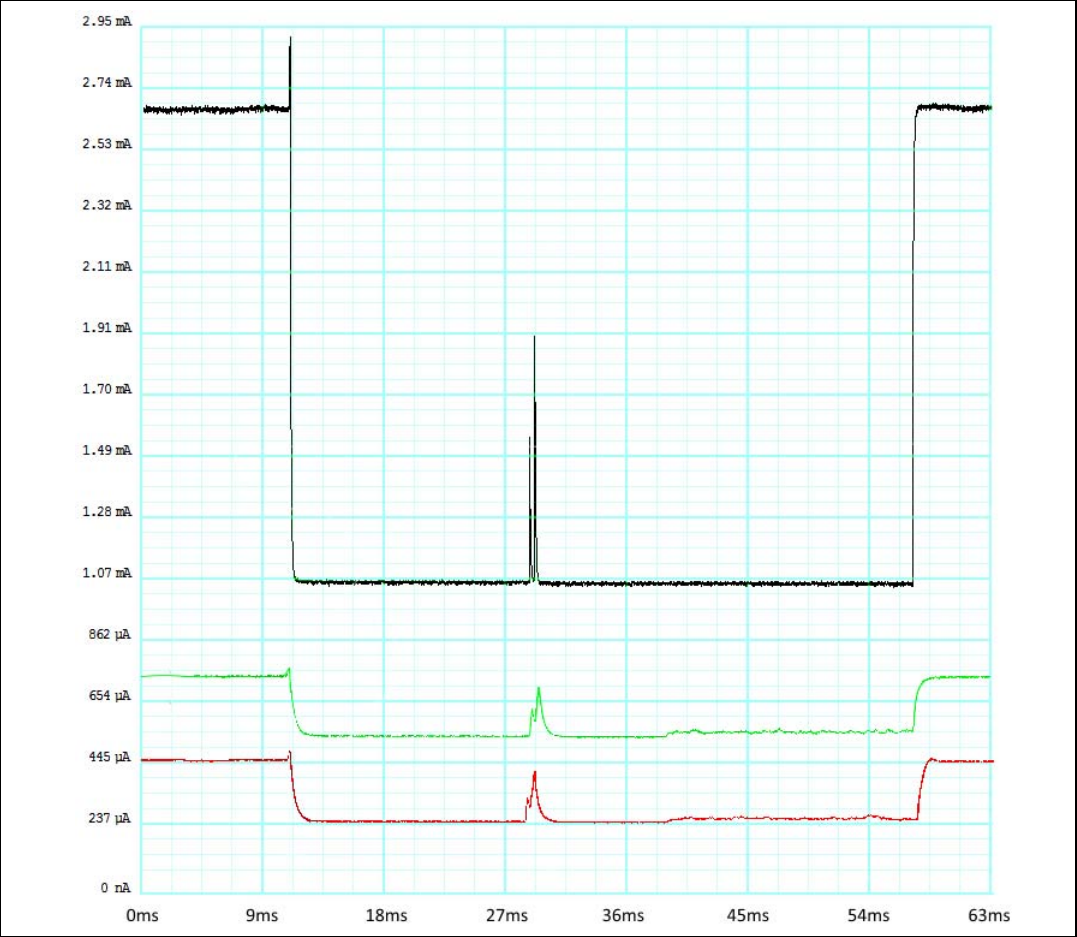


表 20. 配置 - HSI 和 PLL

图 18 中的曲线	黑色	红色	绿色
空闲模式	睡眠		
调试接口	关闭		
平均电流 2、3、4 阶段	1.05 mA	0.536 mA	0.250 mA
时钟	HSI16		使用 LSE 自动校准 MSI ; Sysclk 2 MHz

表 20. 配置 - HSI 和 PLL（续）

图 18 中的曲线	黑色	红色	绿色
调压器	范围 2		低功耗运行
AHB/APB 比	1/1	8/1	1/1
波特率	57600		
使用的 SW	DMA 操作（Com DMA），禁用半缓冲中断		

6.2.6 使用两个振荡器

若要使用更高通信速率的 Stop 模式，必须用 HSI16 源来为 LPUART 外设提供时钟。不必为 Sysclk 重用同一个 16 MHz 时钟，尤其是当速率太高时。对于 57600 波特速率，可以用 MSI 的 2 MHz 频率，以节省大量的能耗。

图 19. 对于 57600 波特速率使用 Stop

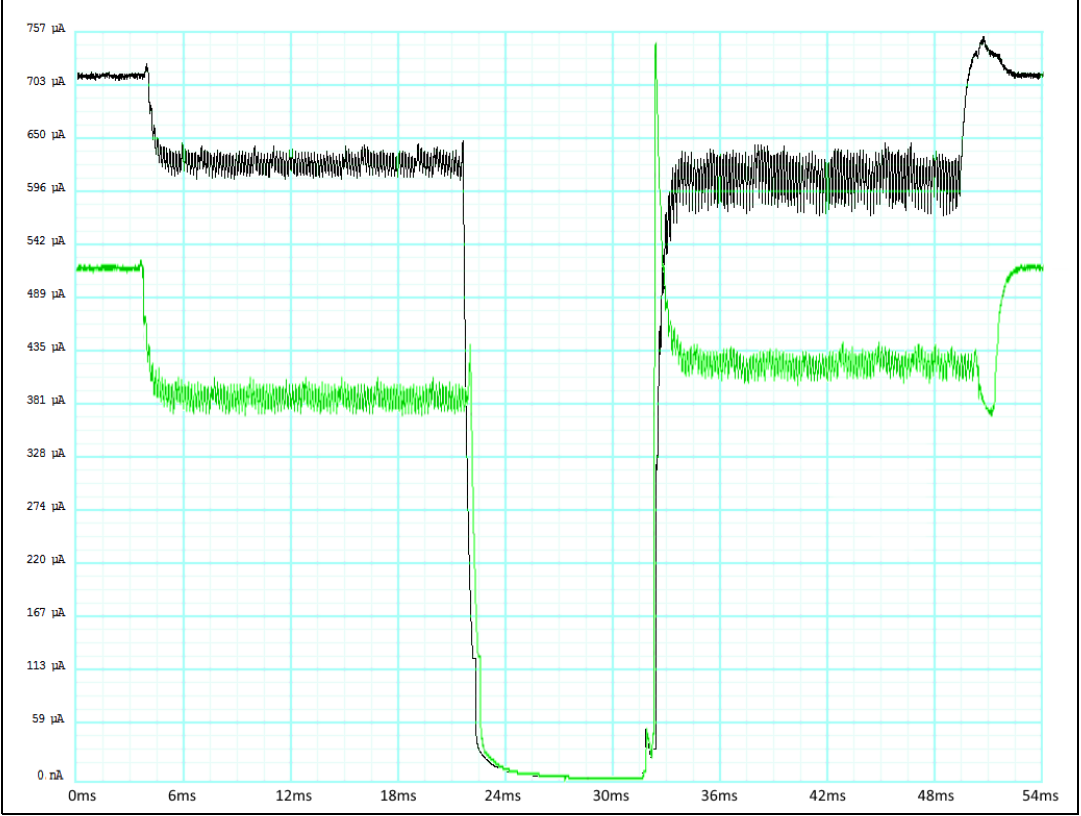


表 21. 配置 - 对于 56700 波特速率使用 Stop

图 19 中的曲线	黑色	绿色
空闲模式	Stop 2	
调试接口	关闭	
平均电流 2、3、4 阶段	480 µA	321 µA

表 21. 配置 - 对于 56700 波特速率使用 Stop（续）

图 19 中的曲线	黑色	绿色
时钟	HSI16	使用 LSE 自动校准 MSI ； Sysclk 2 MHz
调压器	范围 2	低功耗运行
AHB/APB 比	8/1	1/1
波特率	57600	
使用的 SW	IT 操作（Com IT）， Rx Stop	IT 操作（Com IT）

注： 对于 Stop 2 模式，必须退出 LPRUN。

7 示例项目

可下载软件包 X-CUBE-LPUART 中提供了一个固件示例，其目的为示范本文中提到的不同配置。多数配置可通过使用 *main.c* 源文件中的定义部分开关。

需要两个板子来重现这些测量，称为主板和重复板，它们加载了由 BOARD2 选项生成的 DMA FW。

应在主板而不是重复板测量功耗。

本例已经使用 STM32L0 和 STM32L4 Nucleo 板开发并测试。

7.1 硬件设置

只需使用两根线交叉连接 Tx 引脚和 Rx 引脚。可使用 JP6 引脚监测功耗。可以组合使用 STM32L4 和 STM32L0 Nucleo 板，只要记住它们的 Rx 和 Tx 引脚位置是对调的就行。

7.2 配置示例

示例固件的各种特性和工作模式都可使用下列预处理宏定义配置：

DEBUG_OFF – 禁用端口 A 上的 SWD 调试能力及误差检查

BOARD2 – 将软件配置为重复板（仅存在于 DMA 版本）

UI – 启用用户界面 – 启用了按钮和 LED 后，将无法正确读取功耗

BD_SPEED – 通信波特率设置

LPRUN – 配置低功耗运行模式（对于 L0，仅在轮询版本中）

PWR_CR_VOS_CONF – 主功率调压器配置（默认为范围 3）

TXSLEEP – 配置发送字节间的 Sleep 模式

RXSLEEP – 配置接收字节间的 Sleep 模式

RXSTOP – 配置接收字节间的 Stop 模式

HSI – 将系统时钟设置为 4MHz HSI 模式。

RCC_MSIRANGE_SET – 对于禁用 HSI 的情况，设置 MSI 速率范围。

7.3 示例操作

禁用 UI 时，板子无限循环地交换消息。在通信错误及启用调试时，停止循环。

开启 UI 选项时，板子会等待用户按下按钮开始循环，点亮 LED 表示激活，但是所有这些都会对电流消耗有很大影响，因此对于所有功率测量都应禁用 UI。

8 结论

并没有一个最佳方法来配置 LP UART 达到可能的最低功耗。

取决于应用和操作限制，必须考虑很多参数，总的指导意见总结如下：

- 使用较低频率；
- 使用较低电压；
- 尽可能将 MCU 保持在低功耗模式。

优化软件对于整个低功耗解决方案也至关重要。

9 版本历史

表 22. 文档版本历史

日期	版本	变更
2015 年 4 月 22 日	1	初始版本。
2015 年 5 月 26 日	2	更新了 前言 ，加入 X-CUBE-LPUART 软件。 更新了 第 7 节：示例项目 。
2015 年 10 月 12 日	3	引入了 STM32L4 系列。 更新了 前言 、 第 2.1.1 节：时钟子系统 、 第 2.2 节：与 USART 外设的比较 、 第 4.2 节：GPIO 配置 、 第 4.4.1 节：使用 Stop 和 Sleep 模式 、 第 4.4.2 节：运行时配置 、 第 5.2 节：丢字节 、 第 7 节：示例项目 及其子章节。 更新了表 1 至 15。 更新了 图 8：三种不同的外设时钟分频设置 和 图 12：调压器设置比较 。 增加了 第 2.1 节：L0 和 L4 系列比较 及其子章节、 第 4.1 节：从 SRAM 执行 、 第 6.2 节：STM32L476 Nucleo 板上的测量 及其子章节。

表 23. 中文文档版本历史

日期	版本	变更
2016 年 6 月 3 日	1	中文初始版本。

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和 / 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。

© 2016 STMicroelectronics - 保留所有权利 2016