

1 介绍

本应用笔记概括介绍了测量外设功耗的步骤，并指出了测量中的一些关键点。该文档以低功耗定时器 (LPTMR) 和 LPUART 为例，介绍了测量外围设备功耗的方法。测试代码是在 IAR 和 FRDM-KE15Z 开发板上开发的。

用于 Kinetis® MCU 的**功耗估算工具**使您可以通过简单的图形界面快速估算和优化系统的功耗。该工具使用应用笔记中描述的方法进行数据测量。

2 测量方式

2.1 状态定义和计算方法

估算外设功耗的基本思想是测量外设的未初始化状态和工作状态之间的电流差。当前的差异称为外设加法器。但是，某些因素会影响测量的准确性。例如 CPU 内核的动态功耗、功耗模式设置、电流表的精度和范围等等。因此，必须找到一些方法来解决这些问题并提高测量精度。

为了避免将 CPU 动态功耗添加到外围加法器中，在测量电流时需要停止 CPU 内核运行。因此，CPU 需处于停止状态的 Wait/Stop 电源模式以用于测量电流。更具体地说，使用哪种功耗模式 (Wait/Stop) 取决于外围设备的电流消耗。如果外围设备的电流消耗值约为 nA 级并且外设可以在 Stop 模式下工作，则建议使用 Stop 模式。在测量通信外设 LPUART, LPSP1, LPI2C 等时，需要传输数据，请使用 DMA 而非 CPU。这意味着内存和外设之间的数据传输是由 DMA 处理的，而无需 CPU 的交互。

根据以上考虑，定义用于获取外设加法器的两个电源状态 — BASELINE 和 RUN。

- BASELINE：设置时钟源，功能时钟，电源模式，然后根据需要初始化 DMA 或其他外设。在测量过程中，令芯片进入 Wait/Stop 模式。在基准条件下，不启用外设时钟门。仅测量功耗模式和时钟配置的平均电流。
- RUN：启用外设时钟门，初始化外设并启动外设。例如：根据外围设备传输数据、执行 A/D 转换等。然后，令芯片进入 Wait/Stop 电源模式。

IDD_{BASELINE} 测量完成后，需要一个中断信号在 Wait/Stop 状态下唤醒芯片，并将电源状态从 BASELINE 变为 RUN。板载按键 SW2 用于触发中断。测量 IDD_{RUN} 之后，可以使用以下公式估算外围设备的功耗：

$$Adder = IDD_{RUN} - IDD_{BASELINE}$$

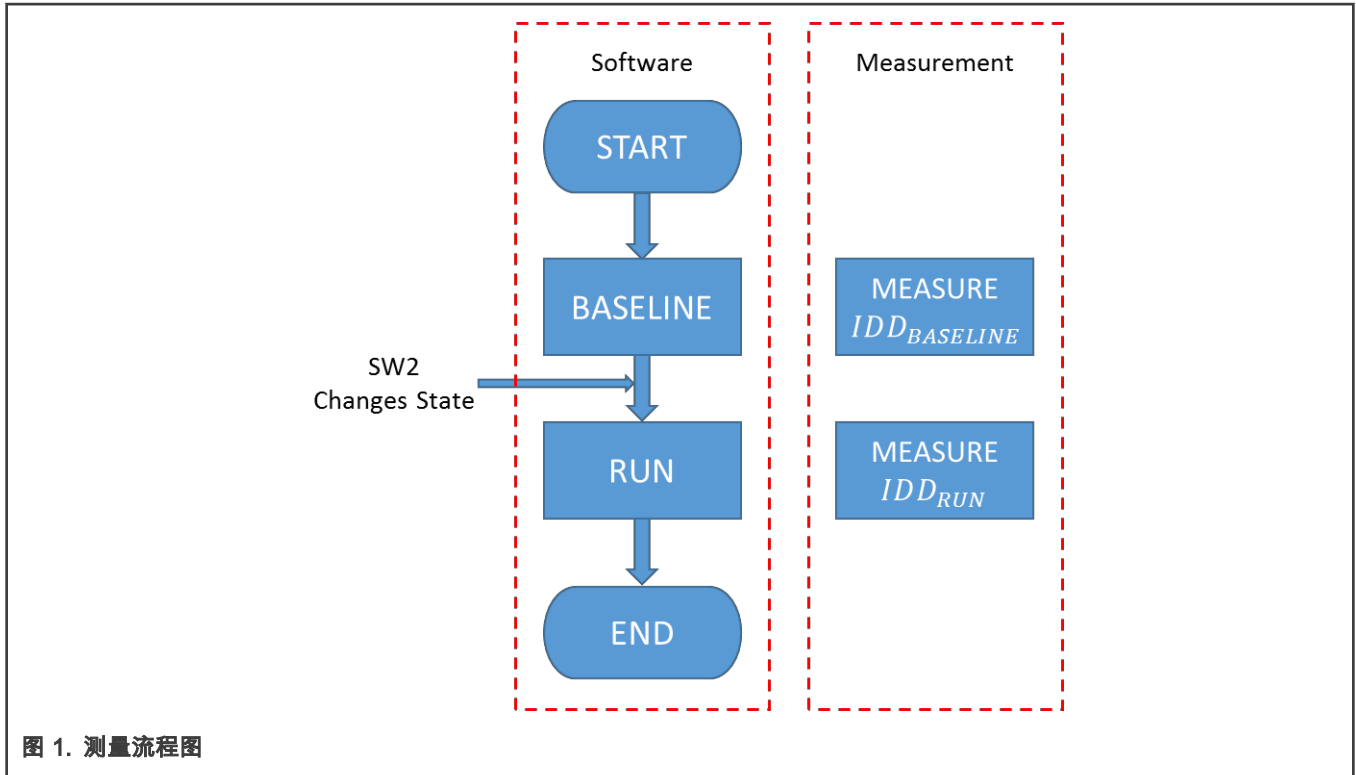
Equation 1.

您需要选择合适的电流表范围。如果状态从 BASELINE 变为 RUN，且电源模式为 Stop 模式，则可能需要更改电流表的范围。否则，当按下 SW2 时电流值可能会超出电流表范围，这可能会导致意外情况发生，例如复位。图 1 显示了整个测量过程。

目录

1	介绍.....	1
2	测量方式.....	1
2.1	状态定义和计算方法.....	1
2.2	硬件设置.....	2
3	示例：如何测量 LPTMR 的功耗.....	3
4	示例：如何测量 LPUART 的功耗....	4
5	结论.....	6
6	修订记录.....	6





2.2 硬件设置

FRDM-KE15Z 开发板设置：

- 电源：电源是通过 PC USB 端口到 SDA USB 的 5 V 直流电。板载 LDO 可以提供 3.3 V 直流电。本应用笔记的示例选择 5 V 作为电源，因此需要连接 J15 的 1-2。
- 电流测量：带有电流表的 J14 可以用于测量每种状态的 IDD ($VDD + VDDA$)。
- 状态更改：板上按键 SW2 用于更改状态 BASELINE -> RUN。

注意

在程序运行之前，请除去 J7 和 J8 的跳线帽，因为 Open-SDA 电路会导致更多的功耗。

如果选择了 Stop 模式，则下载后需要重新为 FRDM-KE15Z 板通电。

对于通信外围设备，当工作模式为 Transmit 时，板载走线长度对外设功耗 (Adder) 影响很大。

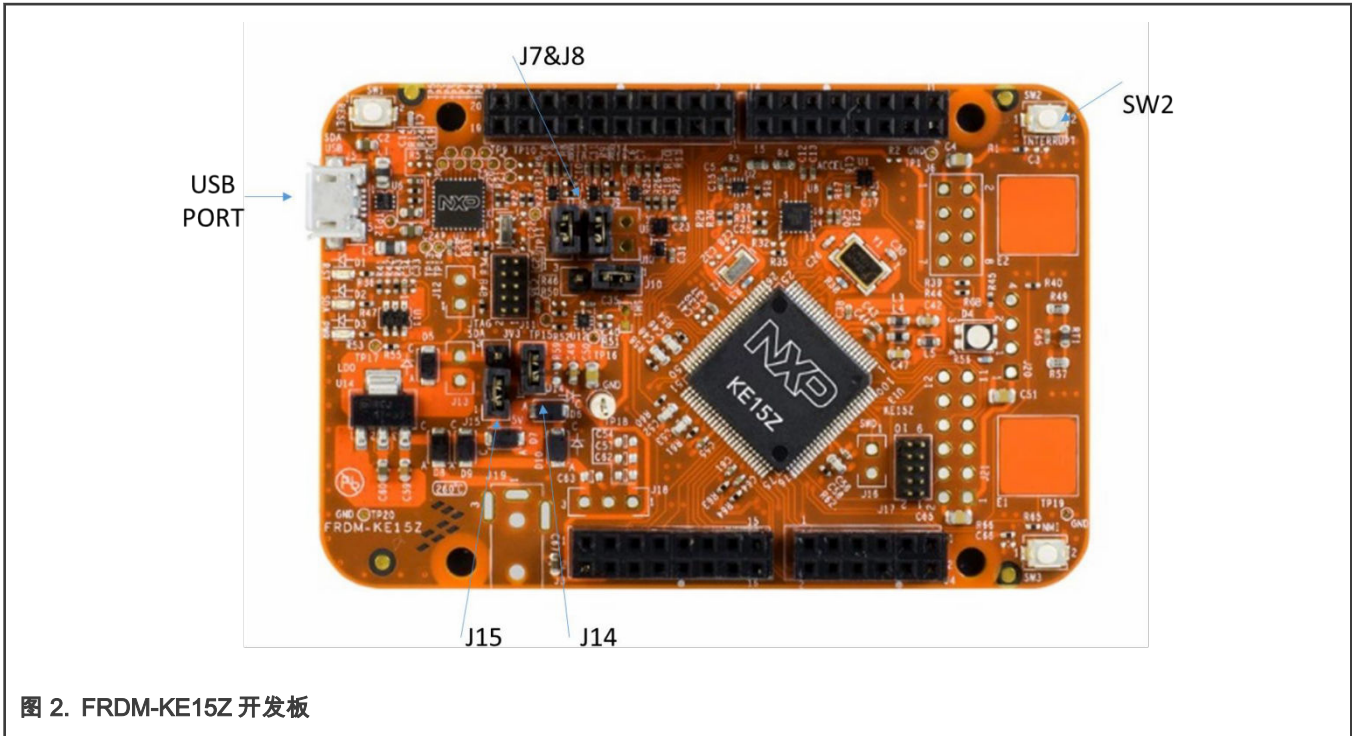


图 2. FRDM-KE15Z 开发板

3 示例：如何测量 LPTMR 的功耗

您可以将 LPTMR 配置为在所有电源模式下带可选预分频器的计时器或带有可选毛刺滤波器的脉冲计数器。在此示例中，LPTMR 被配置为时间计数器，并列出了测量 LPTMR 功耗的步骤。

步骤如下：

1. 在此步骤中确定时钟源和功能时钟的频率。不同的配置会产生不同的功耗。您可以根据自己的要求进行配置。在此示例中，选择 FIRC 48 MHz 作为时钟源，功能时钟的频率为 48 MHz。板载键 SW2 需要配置为 GPIO 上拉并在下降沿中断。因为外设的电流消耗值约为 nA 级，所以需要选择 Stop 模式。

```

BASELINE ()
{

    /****** BASELINE State *****/
    lptmr_config_t lptmrConfig;
    port_pin_config_t pinConfig = {0};
    BOARD_BootClockRUN();

    /* Initialize the on-board key SW2 */
    CLOCK_EnableClock(kCLOCK_PortB);
    pinConfig.pullSelect = kPORT_PullUp;
    pinConfig.mux = kPORT_MuxAsGpio;
    PORT_SetPinConfig(BOARD_SW2_PORT, BOARD_SW2_GPIO_PIN, &pinConfig);
    PORT_SetPinInterruptConfig(BOARD_SW2_PORT, \
    BOARD_SW2_GPIO_PIN, kPORT_InterruptFallingEdge);
    EnableIRQ(BOARD_SW2_IRQ);

    /* Configure FIRC as Clock Source and Function Clock is 48 MHz */
    clocks_run_48M_firc(1);
    SCG->FIRCCSR = 0x03;    //Enable FIRC and Enable Stop Mode

    /* BASELINE: Enter Stop Power Mode and Measure IDD_BASELINE here */

```

```

SMC_SetPowerModeStop(SMC, kSMC_PartialStop);
}

```

2. 用电流表测量电流消耗 $IDD_{BASELINE}$ 。然后按 SW2 键，将状态从 BASELINE 更改为 RUN。在按下 SW2 之前，可能需要更改电流表的范围。
3. 选择定时器模式，配置定时器周期并启用定时器。LPTMR 的中断不应启用，因为该中断将导致电源模式从 Stop 变为 Run。使芯片再次进入 Stop 模式。

```

RUN()
{
/***** RUN State *****/
    /* Initialize LPTMR */
    CLOCK_SetIpSrc(kCLOCK_Lptmr0, kCLOCK_IpSrcFircAsync);
    LPTMR_GetDefaultConfig(&lptmrConfig);
    lptmrConfig.timerMode = kLPTMR_TimerModeTimeCounter;
    lptmrConfig.prescalerClockSource = kLPTMR_PrescalerClock_0;
    lptmrConfig.enableFreeRunning = false;
    lptmrConfig.bypassPrescaler = false;
    lptmrConfig.value = kLPTMR_Prescale_Glitch_11;    LPTMR_Init(LPTMR0, &lptmrConfig);

    /* Set timer period, as LPTMR counter is 16-bit only, the clock source must be prescaled
    4096 */
    LPTMR_SetTimerPeriod(LPTMR0, USEC_TO_COUNT(LPTMR_USEC_COUNT, \
        (CLOCK_GetFreq(kCLOCK_ScgFircAsyncDiv2Clk)/4096));

    /* Start counting */
    LPTMR_StartTimer(LPTMR0);

    /* RUN: Enter Stop Power Mode and Measure IDD_RUN here */
    SMC_SetPowerModeStop(SMC, kSMC_PartialStop);
}

```

4. 用电流表测量电流消耗 IDD_{RUN} 。现在就可以估计 LPTMR 的功耗 (Adder)。

注意

如果选择了 Stop 模式，则 FRDM-KE15Z 开发板需要在下载后重新供电。

4 示例：如何测量 LPUART 的功耗

低功耗通用异步收发器 (LPUART) 是一种广泛使用的通信外围设备。对于通信外围设备，板载走线长度对功耗有很大的影响。本示例列出了测量 LPUART 功耗的步骤。通常，通信外围设备需要两块板进行测量：一个是发送器，另一个是接收器。

1. 在此步骤中确定时钟源和功能时钟的频率。不同的配置会产生不同的功耗。您可以根据自己的要求进行配置。在此示例中，选择 FIRC 作为时钟源，并且功能时钟的频率为 48 MHz。板载按键 SW2 需要配置为 GPIO 上拉并在下降沿中断。内存和外设之间的数据传输由 DMA 处理，而无需 CPU 的交互。之所以选择 Wait 模式，是因为外围设备的电流消耗值约为 uA 级。

```

BASELINE()
{
/***** BASELINE State *****/
    port_pin_config_t pinConfig = {0};
    edma_config_t edmaConfig;
    edma_transfer_config_t transferConfig;
    lpuart_config_t lpuartConfig;
    clocks_run_48M_firc(2);

    /* Initialize the on-board key SW2 */
    CLOCK_EnableClock(BOARD_SW2_CLOCK);
    pinConfig.pullSelect = kPORT_PullUp;
}

```

```
pinConfig.mux = kPORT_MuxAsGpio;
PORT_SetPinConfig(BOARD_SW2_PORT, BOARD_SW2_GPIO_PIN, &pinConfig);
PORT_SetPinInterruptConfig(BOARD_SW2_PORT, BOARD_SW2_GPIO_PIN, \
kPORT_InterruptFallingEdge);
EnableIRQ(BOARD_SW2_IRQ);

/* Configure FIRC as Clock Source and Function Clock is 48 MHz */
clocks_run_48M_firc(1);

/**
 * DMA/DMAMUX configure
 * DMAMUX configure channel to transmit or receive with transmit/receive module trigger source.
 * DMA configure source, destination address and continuous working
 */
DMAMUX_Init(DMAMUX);
#ifdef LPUART_ADDER_RECEIVER
DMAMUX_SetSource(DMAMUX, 0, 2); // LPUART0 RX
DMAMUX_EnableChannel(DMAMUX, 0);
#else
DMAMUX_SetSource(DMAMUX, 0, 3); // LPUART0 TX
DMAMUX_EnableChannel(DMAMUX, 0);
#endif

// Init EDMA and Channel0
EDMA_GetDefaultConfig(&edmaConfig);
edmaConfig.enableHaltOnError = false;
EDMA_Init(DMA0, &edmaConfig);
EDMA_ResetChannel(DMA0, 0);

// Disable auto stop request and dma aysnc request
EDMA_EnableAsyncRequest(DMA0, 0, true);
EDMA_EnableAutoStopRequest(DMA0, 0, false);

#ifdef LPUART_ADDER_RECEIVER
transferConfig.srcAddr = LPUART_GetDataRegisterAddress(LPUART0);
transferConfig.destAddr = (uint32_t)recvBuf;
#else
transferConfig.srcAddr = (uint32_t)&testData;
transferConfig.destAddr = LPUART_GetDataRegisterAddress(LPUART0);
#endif
// Offset applied to current address to form next transfer address
transferConfig.srcTransferSize = kEDMA_TransferSize1Bytes;
transferConfig.destTransferSize = kEDMA_TransferSize1Bytes;
transferConfig.srcOffset = 0;
#ifdef LPUART_ADDER_RECEIVER
transferConfig.destOffset = 1;
#else
transferConfig.destOffset = 0;
#endif
// 1bytes per DMA request
transferConfig.minorLoopBytes = 1;
transferConfig.majorLoopCounts = 100;

EDMA_SetTransferConfig(DMA0, 0, &transferConfig, NULL);
#ifdef LPUART_ADDER_RECEIVER
DMA0->TCD[0].DLAST_SGA = -sizeof(recvBuf);
#endif

// Start dma channel0 transfer
EDMA_EnableChannelRequest(DMA0, 0);
```

```

/* BASELINE: Enter Wait Power Mode and Measure IDD_BASELINE here */
SMC_SetPowerModeWait(SMC);
}

```

- 用电流表测量电流消耗 $IDD_{BASELINE}$ 。然后按 SW2 键，将状态从 BASELINE 更改为 RUN。
- 在所需的波特率和模式下配置 LPUART，然后使芯片再次进入 WAIT 模式。

```

RUN()
{
    /****** RUN State *****/
    /* Initialize LPUART */
    CLOCK_EnableClock(kCLOCK_PortA);
    PORT_SetPinMux(PORTA, 10U, kPORT_MuxAlt3);
    PORT_SetPinMux(PORTA, 11U, kPORT_MuxAlt3);
    CLOCK_SetIpSrc(kCLOCK_Lpuart0, kCLOCK_IpSrcFircAsync);
    LPUART_GetDefaultConfig(&lpuartConfig);
    lpuartConfig.baudRate_Bps = 460800;
    LPUART_Init(LPUART0, &lpuartConfig, CLOCK_GetFreq(kCLOCK_ScgFircAsyncDiv2Clk));

#ifdef LPUART_ADDER_RECEIVER
    LPUART_EnableRxDMA(LPUART0, true);
    LPUART_EnableRx(LPUART0, true);
#else
    LPUART_EnableTxDMA(LPUART0, true);
    LPUART_EnableTx(LPUART0, true);
#endif

    /* RUN: Enter Wait Power Mode and Measure IDD_RUN here */
    SMC_SetPowerModeWait(SMC);
}

```

- 用电流表测量电流消耗 IDD_{RUN} 。现在可以估算 LPUART 的功耗 (Adder)。

注

对于通信外围设备，当工作模式为“发送”时，板载走线长度会影响外设的功耗 (Adder)。

5 结论

本应用笔记概述了测量外设功耗的步骤，并给出了两个模块功耗的测量方法。其他模块也可以用类似的方式进行测量。合适的电源模式有助于提高测量精度。

6 修订记录

表 1. 修订记录

版本号	日期	说明
0	2017 年 1 月	初始版本

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Limited warranty and liability — Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Right to make changes - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, ICODE, JCOP, LIFE, VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μ Vision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. M, M Mobileye and other Mobileye trademarks or logos appearing herein are trademarks of Mobileye Vision Technologies Ltd. in the United States, the EU and/or other jurisdictions.

© NXP B.V. 2017-2021.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 2017 年 1 月
Document identifier: AN5402

