

AN14332

MCXC444功耗模式的切换应用

第1版—2024年7月9日

应用笔记

文档信息

信息	内容
关键词	AN14332、MCXC444、功耗模式、PMC、WFE
摘要	本应用笔记提供了有关每种电源模式的详细信息，并包含了一些SDK功耗模式切换示例演示中的用例。



1 介绍

MCXC444系列微控制器为功耗敏感型市场提供了超低功耗特性。该MCU系列实现了多种低功耗模式来满足这方面的要求。本应用笔记提供了关于每种功耗模式的详细信息，并包含了一些SDK功耗模式切换示例演示中的用例。还提供了使用每种功耗模式的说明。

MCUXpresso SDK为用户提供了强大的外设驱动程序、协议栈、中间件和示例应用程序，旨在简化和加速基于任意恩智浦MCU的应用程序的开发。MCUXpresso SDK是免费提供的，包含在开源许可下的所有硬件抽象和外设驱动程序软件的完整源代码。

本文重点介绍了电源管理控制器（PMC）、系统模式控制器（SMC）、多用途时钟发生器简化版（MCG-Lite）和低功耗唤醒单元（LLWU）。

2 MCXC444 MCU的功耗模式

本节详细介绍了Arm Cortex-M架构和MCXC444 MCU的功耗模式。

2.1 Arm Cortex-M0+核的基本功耗模式

Arm Cortex-M0+使用Arm Cortex-M架构的以下基本功耗模式：

- 运行（RUN）
- 睡眠（SLEEP）模式：停止处理器时钟
- 深度睡眠（DEEP SLEEP）模式：停止系统时钟，切换锁相环（PLL）和闪存

注： Arm Cortex-M0+ 处理器的睡眠模式可降低功耗。

该系统可能会产生虚假的唤醒事件。例如，调试操作会唤醒处理器。因此，软件必须能够在发生此类事件后将处理器重新置于睡眠模式。程序可以有一个空闲循环，来使处理器返回到睡眠模式。

要进入低功耗模式（睡眠/深度睡眠），请使用以下三条指令通知处理器：

- **等待中断（WFI）**：WFI指令会使处理器立即进入睡眠模式。当处理器执行WFI指令时，会停止执行指令并进入睡眠模式。
- **等待事件（WFE）**：WFE指令使处理器根据I位事件寄存器的值（由SEV指令设置）进入睡眠模式。当处理器执行WFE指令时，会按如下方式检查事件寄存器的值：
 - 0=处理器停止执行指令，并进入睡眠模式。
 - 1=处理器将寄存器设置为0，并继续执行指令，而不进入睡眠模式。
- **发送事件（SEV）**：SEV指令会将一个事件发送给多处理器系统中的所有处理器。它还会设置本地事件寄存器。

在Arm Cortex-M0+核中，SCB寄存器会控制在WFI/WFE指令后处理器进入低功耗模式的行为。

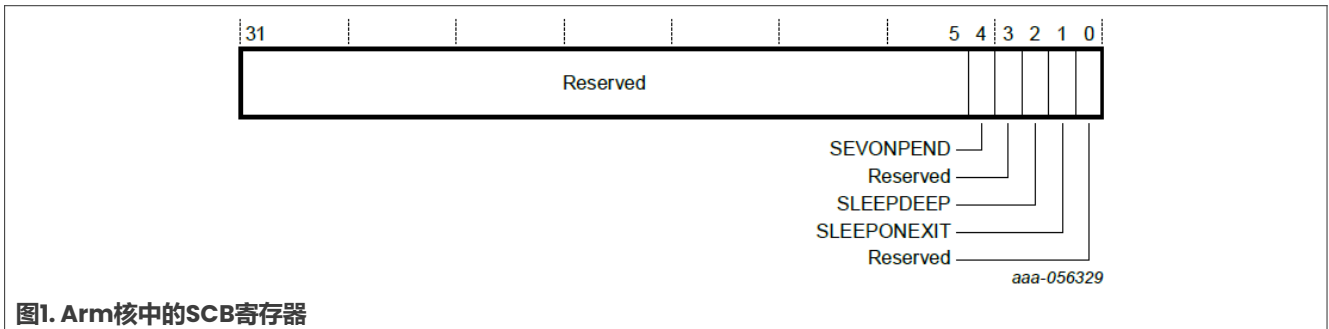


图1. Arm核中的SCB寄存器

图1展示了以下内容：

- SCB[SLEEPDEEP]：该位控制处理器是使用睡眠模式还是深度睡眠模式作为其低功耗模式。
 - 0 =睡眠
 - 1 =深度睡眠
- SCB[SLEEPONEXIT]：该位表示从处理进程（Handler）模式（中断处理进程）返回到主线程（Thread）模式（main()函数）时，是否退出即睡眠：
 - 0 =返回Thread模式时不睡眠，直接返回到main()函数。
 - 1 =在从ISR返回到Thread模式后，再次进入睡眠模式或深度睡眠模式，而不返回到main()函数。将此位设置为1可以使中断驱动的应用程序避免返回到一个空的主应用程序。
- SCB[SEVONPEND]：该位根据挂起位发送事件；
 - 0 =只有使能的中断或事件才能唤醒处理器，禁用的中断除外。
 - 1 =使能的事件和所有中断（包括禁用的中断）都可以唤醒处理器。当事件或中断变为挂起状态时，事件信号会从WFE唤醒处理器。如果处理器没有等待某个事件，则该事件将被注册并影响下一个WFE。处理器也会在执行SEV指令或发生一个外部事件时被唤醒。

此处，WFE作为WFI的轻量级版本，无需恢复上下文即可延迟CPU执行，从而节省了低功耗模式转换期间的周期。WFE指令根据1位事件寄存器的值使处理器进入睡眠模式。当处理器执行WFE指令时，它会检查该事件寄存器的值：

- 0 = 处理器停止执行指令，并进入睡眠模式。
- 1 = 处理器将该寄存器设置为0，并继续执行指令，而不进入睡眠模式。

如果该事件寄存器为“0”，WFE将暂停执行，直到发生了以下事件之一：

- 异常事件，除非被异常掩码寄存器或当前优先级屏蔽。
- 在系统控制寄存器中设置SEVONPEND时，异常进入挂起状态。
- 使能调试时发生“调试进入”请求事件。
- 由外设或多处理器系统中的其他处理器使用SEV指令发出信号的事件。

2.2 MCXC444中的扩展功耗模式

在MCXC444中，其内核使用WFI指令来进入睡眠和深度睡眠模式。它还扩展了功耗模式及其关系，如表1所示。

表1. MCXC444 MCU的功耗模式

Arm CM0+的功耗模式	MCXC444 MCU的功耗模式	唤醒模块	复位
运行	运行、VLPR	—	—
运行	CPO	AWIC/NVIC	否
睡眠	等待 (WAIT)、VLPW	NVIC	否
深度睡眠	停止 (STOP)、VLPS	WIC	否
深度睡眠	PSTOPI	AWIC	否
深度睡眠	PSTOP2	AWIC/NVIC	否
深度睡眠	LLS	LLWU	否
深度睡眠	VLLSx (x=0/1/3)	LLWU	是

表1中列出的唤醒模块的含义如下所示：

- NVIC：任何中断源都可以将MCU从等待/VLPW模式唤醒。
- AWIC：只有参考手册中的AWIC唤醒源才能将MCU从停止/VLPS模式唤醒。
- LLWU：只有参考手册中的LLWU唤醒源才能将MCU从LLS/VLLSx模式唤醒。要从VLLSx模式唤醒MCU，请执行复位流程并调用LLWU复位。
- 对于计算操作（CPO）模式，Arm核处于运行模式。任何异步中断和Arm核同步中断都可以唤醒在运行模式下的MCU。

表2所示为每种功耗模式的详细说明。

表2. 功耗模式说明

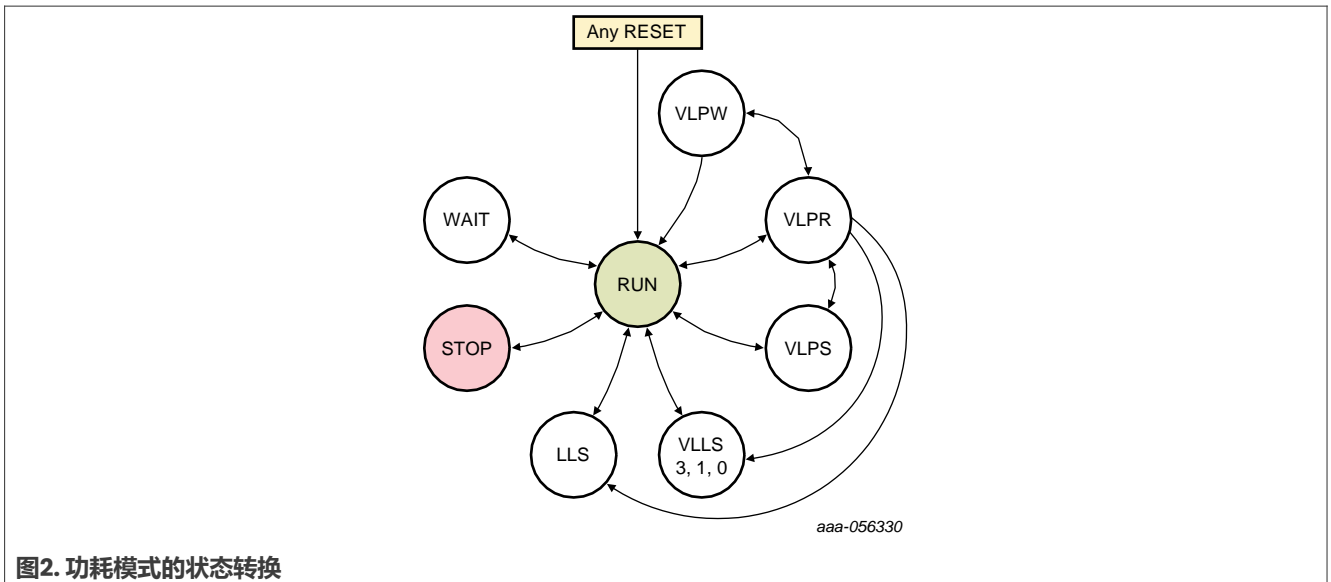
模式	说明
运行	MCU可以全速运行，内部电源完全可调节，即运行调节。此模式也称为正常运行模式。
等待	<ul style="list-style-type: none"> • 内核时钟被门控关闭。 • 系统时钟继续运行。 • 总线时钟（如启用）继续运行。 • 保持运行调节。
停止	<ul style="list-style-type: none"> • 内核时钟被门控关闭。 • 在来自支持外设的所有停止应答信号有效后，其他主设备的系统时钟和总线时钟被门控关闭。
VLPR	在此模式下，内核、系统、总线和闪存时钟的最大频率受到限制。有关最大允许频率的详细信息，请参阅《MCXC444子系列参考手册》（文档MCXC444RM）中的“电源管理”章节。
VLPW	<ul style="list-style-type: none"> • 内核时钟被门控关闭。 • 系统、总线和闪存时钟继续运行，但它们的最大频率受到限制。 有关最大允许频率的详细信息，请参阅《MCXC444子系列参考手册》（文档MCXC444RM）中的“电源管理”章节。
VLPS	<ul style="list-style-type: none"> • 内核时钟被门控关闭。 • 在来自支持外设的所有停止应答信号有效后，其他主设备的系统时钟和总线时钟被门控关闭。
LLS	<ul style="list-style-type: none"> • 内核时钟被门控关闭。 • 在来自支持外设的所有停止应答信号有效后，其他主设备的系统时钟和总线时钟被门控关闭。

表2. 功耗模式说明 (续)

模式	说明
	<ul style="list-style-type: none"> 通过降低内部逻辑的电压，将MCU置于低功耗模式。 保留所有系统RAM的内容、内部逻辑和I/O口状态。
VLLS3	<ul style="list-style-type: none"> 内核时钟被门控关闭。 在来自支持外设的所有停止应答信号有效后，其他主设备的系统时钟和总线时钟被门控关闭。 通过关闭内部逻辑，将MCU置于低功耗模式。 保留所有系统RAM的内容并保持I/O口状态。 不保留内部逻辑状态。
VLLS1	<ul style="list-style-type: none"> 内核时钟被门控关闭。 在来自支持外设的所有停止应答信号有效后，其他主设备的系统时钟和总线时钟被门控关闭。 通过关闭内部逻辑和所有系统RAM的电源，将MCU置于低功耗模式。 保持I/O口状态。 不保留内部逻辑状态。
VLLS0	<ul style="list-style-type: none"> 内核时钟被门控关闭。 在支持外设的所有停止应答信号有效后，到其他主设备的系统时钟和总线时钟被门控关闭。 通过关闭内部逻辑和所有系统RAM的电源，将MCU置于低功耗模式。 保持I/O口状态。 不保留内部逻辑状态。 1 kHz LPO时钟被禁用，可以选择使用STOPCTRL[PORPO]启用上电复位 (POR) 电路。

对于MCXC444系列芯片，NMI引脚可以唤醒所有功耗模式。如果总线时钟没有对复位引脚进行滤波，复位引脚会将MCU功耗模式重置为默认运行模式。

图2所示为该芯片可用功耗模式的状态转换。任何复位都会使MCU恢复到正常运行状态。



要从运行/VLPR模式进入目标功耗模式，请执行以下步骤：

1. 要忽略在超低功耗模式下LDO降低电源电压时的警告，请禁用电源管理控制器（PMC）模块中的低压检测（LVD）功能。
2. 要在目标低功耗模式下节省功耗，请禁用不必要的模块/引脚，并设置时钟唤醒源模块以触发低功耗退出事件。
3. 为确保继续工作的模块仍能以可用的时钟源工作，请设置目标模式的时钟源。
4. 解锁SMC -> PMPROT（功耗模式保护）寄存器中所指示的功耗模式，以便目标功耗模式可进入。
5. 在SMC -> PMCTRL（功耗模式控制）寄存器和SMC > STOPCTRL（停止控制）寄存器中设置目标模式。
6. 要进入低功耗模式，请调用WFI或WFE。

要退出低功耗模式，请执行以下步骤：

1. 等待唤醒事件触发预置的唤醒源。
2. 对于VLLSx模式，LLWU专门设计作为一个唤醒模块，用于收集所有可用的唤醒源。

对于某些复位后的唤醒例程，应清除PMC -> REGSC[ACKISO]位以解锁端口引脚，这些引脚在某些超低功耗模式下被锁定并保持稳定。

表3所示为低功耗模式（LLS和VLLSx）下的模块操作。

表3. 低功耗模式下的模块操作

模块	VLPR	VLPW	停止	VLPS	LLS	VLLSx
内核模块						
NVIC	FF	FF	静态	静态	静态	OFF
系统模块						
模式控制器	FF	FF	FF	FF	FF	FF
LLWU	静态	静态	静态	静态	FF	FF
稳压器	低功耗	低功耗	ON	低功耗	低功耗	<ul style="list-style-type: none"> 在VLLS3中低功耗 在VLLS0/1中关闭
断电检测	ON	ON	ON	ON	ON	<ul style="list-style-type: none"> 在VLLS1/3中开启 在VLLS0中可选禁用
时钟						
1 kHz LPO	ON	ON	ON	ON	ON	<ul style="list-style-type: none"> 在VLLS1/3中开启 在VLLS0中关闭
系统振荡器 (OSC)	<ul style="list-style-type: none"> OSCERCLK 最大16MHz晶振 	<ul style="list-style-type: none"> OSCERCLK 最大16MHz晶振 	<ul style="list-style-type: none"> OSCERCLK 可选 	<ul style="list-style-type: none"> OSCERCLK 最大16MHz晶振 	OSCERCLK 最大16MHz晶振	<ul style="list-style-type: none"> OSCERCLK 在VLLS1/3中最大16MHz晶振 在VLLS0中关闭
存储器和存储器接口系统模块						

表3. 低功耗模式下的模块操作 (续)

模块	VLPR	VLPW	停止	VLPS	LLS	VLLSx
SRAM_U和SRAM_L	低功耗	低功耗	低功耗	低功耗	低功耗	<ul style="list-style-type: none"> 在VLLS3中低功耗 在VLLS0/1中关闭
系统寄存器文件	上电	上电	上电	上电	上电	上电
定时器						
LPTMR	FF	FF	<ul style="list-style-type: none"> 异步操作 PSTOP2中FF 	异步操作	异步操作	异步操作
RTC	<ul style="list-style-type: none"> FF CPO中异步操作 	FF	<ul style="list-style-type: none"> 异步操作 PSTOP2中FF 	异步操作	异步操作	异步操作
人机界面						
段式LCD	<ul style="list-style-type: none"> FF CPO中异步操作 	FF	<ul style="list-style-type: none"> 异步操作 PSTOP2中FF 	异步操作	异步操作	<ul style="list-style-type: none"> 异步操作 VLLS0中关闭

3 测量各种功耗模式下的电流

在本文档中，设计了一款应用软件来测量MCXC444 MCU在各种功耗模式下运行时的电流。采用FRDM-MCXC444开发板作为主要的硬件平台。开发板上的两个按钮用在SLCD屏幕上的切换目标功耗模式的选择。

3.1 开发板设置

FRDM-MCXC444开发板具有可供应用程序使用的测量插座。

JP1是预留的测量插座，如图3和图4所示。

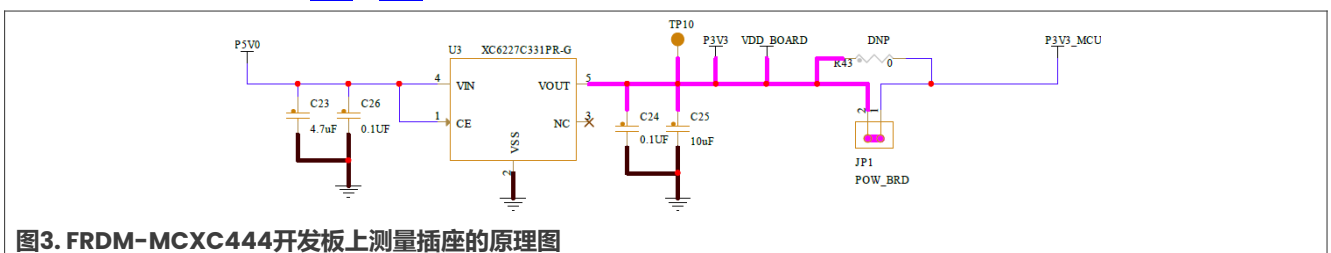


图3. FRDM-MCXC444开发板上测量插座的原理图

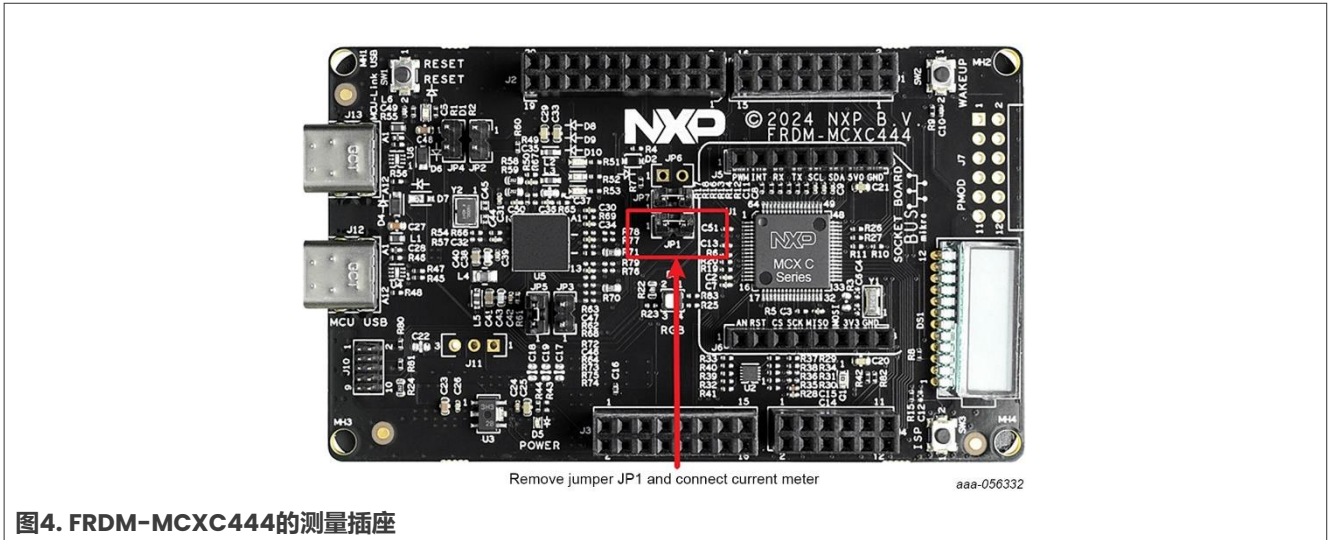


图4. FRDM-MCXC444的测量插座

SLCD在运行过程中显示当前的功耗模式，如图5所示。

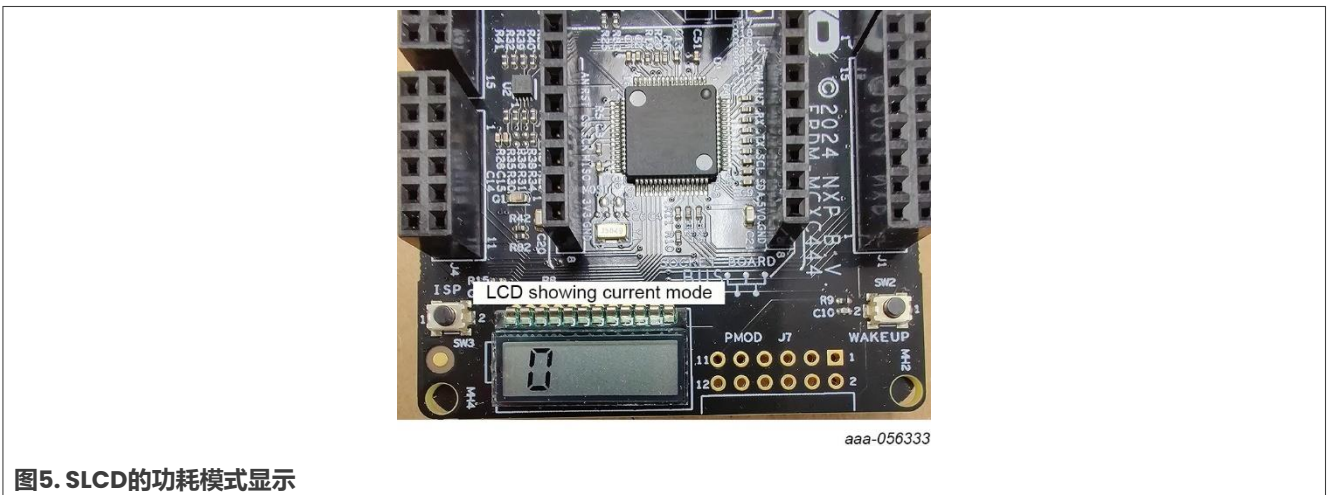


图5. SLCD的功耗模式显示

3.2 软件设计

MCUXpresso SDK软件包提供了SMC、PMC、LLWU和时钟模块的驱动程序。在应用程序软件中，这些驱动程序API可用于与其他外设驱动程序一起操作功耗模式。

3.2.1 通过软件切换功耗模式

在此应用程序演示中，共涵盖了10种功耗模式，如下所示：

```

/* Power mode definition used in application. */
typedef enum _app_power_mode
{
    kAPP_PowerModeRun, /* Normal RUN mode */
    kAPP_PowerModeWait, /* WAIT mode. */
    kAPP_PowerModeStop, /* STOP mode. */
    kAPP_PowerModeVlpr, /* VLPR mode. */
    kAPP_PowerModeVlpw, /* VLPW mode. */
    kAPP_PowerModeVlps, /* VLPS mode. */
    kAPP_PowerModeLls, /* LLS mode. */
    kAPP_PowerModeVlls0, /* VLLS0 mode. */
};

```



```
kAPP_PowerModeVlls1, /* VLLS1 mode. */
kAPP_PowerModeVlls3, /* VLLS3 mode. */
kAPP_PowerModeMax
} app_power_mode_t;
```

- APP_PowerModeSwitch() 函数是执行功耗模式切换的最重要的函数。
- 要控制目标功耗模式，可使用驱动程序的SMC API：

```
void APP_PowerModeSwitch(smc_power_state_t curPowerState, app_power_mode_t
targetPowerMode)
{
    smc_power_mode_vlls_config_t vlls_config;
    vlls_config.enablePorDetectInVlls0 = true;
    switch (targetPowerMode)
    {
        case kAPP_PowerModeVlpr:
            APP_SetClockVlpr(); /* setup the lower clock source for VLPR. */
            SMC_SetPowerModeVlpr(SMC);
            while (kSMC_PowerStateVlpr != SMC_GetPowerModeState(SMC))
            {
            }
            break;
        case kAPP_PowerModeRun:
            /* Power mode change. */
            SMC_SetPowerModeRun(SMC);
            while (kSMC_PowerStateRun != SMC_GetPowerModeState(SMC))
            {
            }
            /* If enter RUN from VLPR, change clock after the power mode
change.
*/
            if (kSMC_PowerStateVlpr == curPowerState)
            {
                APP_SetClockRunFromVlpr(); /* setup the higher clock source for
RUN. */
            }
            break;
        case kAPP_PowerModeWait:
            SMC_PreEnterWaitModes();
            SMC_SetPowerModeWait(SMC);
            SMC_PostExitWaitModes();
            break;
        case kAPP_PowerModeStop:
            SMC_PreEnterStopModes();
            SMC_SetPowerModeStop(SMC, kSMC_PartialStop);
            SMC_PostExitStopModes();
            break;
        case kAPP_PowerModeVlps:
            SMC_PreEnterWaitModes();
            SMC_SetPowerModeVlps(SMC);
            SMC_PostExitWaitModes();
            break;
        case kAPP_PowerModeVlps:
            SMC_PreEnterStopModes();
            SMC_SetPowerModeVlps(SMC);
            SMC_PostExitStopModes();
            break;
        case kAPP_PowerModeLls:
            SMC_PreEnterStopModes();
```

```
        SMC_SetPowerModeVlls(SMC);
        SMC_PostExitStopModes();
        break;
    case kAPP_PowerModeVlls0:
        vlls_config.subMode = kSMC_StopSub0;
        SMC_PreEnterStopModes();
        SMC_SetPowerModeVlls(SMC, &vlls_config);
        SMC_PostExitStopModes();
        break;
    case kAPP_PowerModeVlls1:
        vlls_config.subMode = kSMC_StopSub1;
        SMC_PreEnterStopModes();
        SMC_SetPowerModeVlls(SMC, &vlls_config);
        SMC_PostExitStopModes();
        break;
    case kAPP_PowerModeVlls3:
        vlls_config.subMode = kSMC_StopSub3;
        SMC_PreEnterStopModes();
        SMC_SetPowerModeVlls(SMC, &vlls_config);
        SMC_PostExitStopModes();
        break;
    default:
        break;
}
}
```

- 进入一个新功耗模式类似于在操作系统中切换到具有新工作条件的新任务。
- 要关闭当前的模式并准备进入新模式，在进入之前应执行所需的操作。在新模式下，执行必要的初始化工作。
- 因此，在应用程序演示代码中，创建了APP_PowerPreSwitchHook()和APP_PowerPostSwitchHook()函数来打包这些操作。
- 为了尽量减少MCU的功耗，在进入等待/停止模式之前应禁用不必要的外设，并在唤醒后恢复。
- 在应用程序演示中，用于终端交互的UART外设以及输出引脚在低功耗模式下被禁用。
- 在MCU处于睡眠模式期间，只有OSC32时钟驱动的SLCD和NVIC/AWIC仍然工作。

3.2.2 在低功耗模式下保留SRAM的内容

为了显示SRAM的内容是否可以在VLLSx模式下保留，内部写入了一个软件令牌的变量来指示其内容是否丢失。

1. 在应用程序演示中，进入VLLSx模式之前，将APP_SRAM_PRESERVE_FLAG令牌写入s_app_persist_flag变量。
2. 当MCU再次唤醒后读取该令牌。
3. 当MCU从复位状态唤醒时，软件读取令牌变量并将其与预期值进行比较。
4. 通过SLCD屏幕上的显示和UART终端通知用户。

```
#define APP_SRAM_PRESERVE_FLAG 0x55555555

static volatile uint32_t s_app_persist_flag_attribute__((section(".noinit")));

...

/*!
```

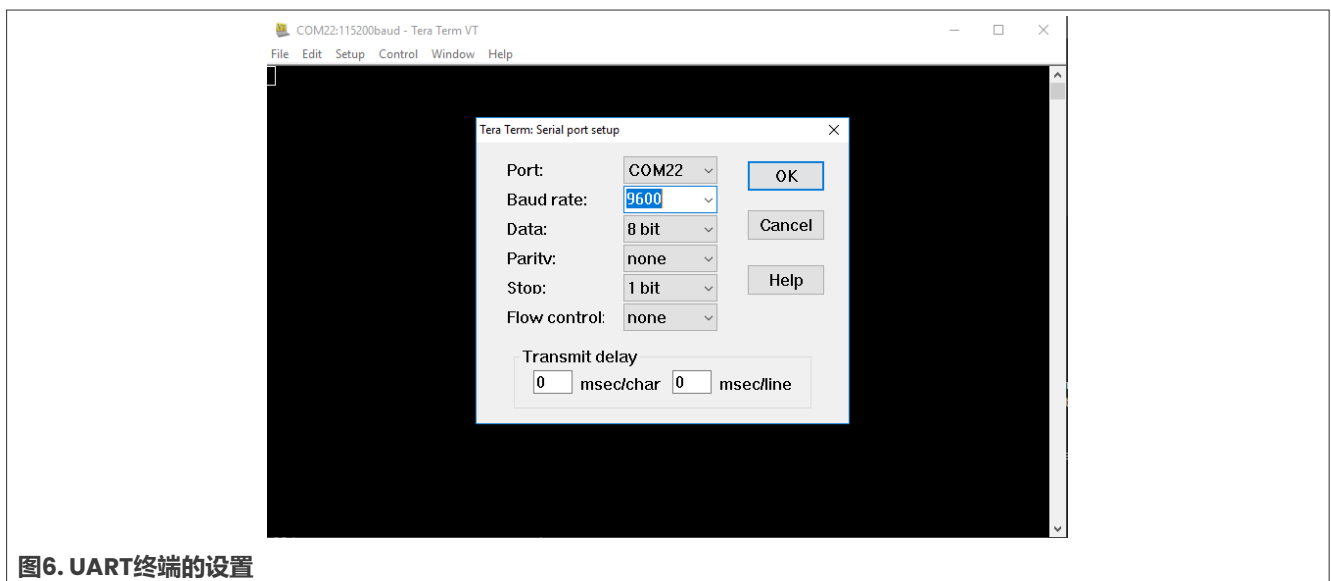
```
* @brief main demo function.
*/
int main(void) {
...
    if (kRCM_SourceWakeup & RCM_GetPreviousResetSources(RCM)) /* Wakeup from
VLLS. */
    {
        PRINTF("\r\nMCU wakeup from VLLS modes...\r\n");

        if (s_app_persist_flag == APP_SRAM_PRESERVE_FLAG)
        { PRINTF("SRAM content preserved...\r\n");
          SLCD_Engine_Show_Num(&s_lcdEngine, 1, 2, true);
        } else {
          PRINTF("SRAM content not preserved...\r\n");
          SLCD_Engine_Show_Num(&s_lcdEngine, 0, 2, true);
        }
    }
...
}
```

3.3 运行应用程序演示工程

构建工程并将镜像文件下载到FRDM-MCXC444开发板后，运行应用程序演示以测量不同功耗模式下的工作电流。UART终端可以输出日志信息。

1. 关于万用表。如前所述，将万用表（电流测量模式）串联接入J20中。
2. 将板载调试器连接到PC，打开用于UART通信的终端工具，配置如图6所示：
 - 波特率 = 9600
 - 8个数据位
 - 无奇偶校验位



3. 现在，应用程序演示开始运行。
4. 初始功耗模式为运行模式。对于运行模式，SLCD显示“0”，如图7所示。

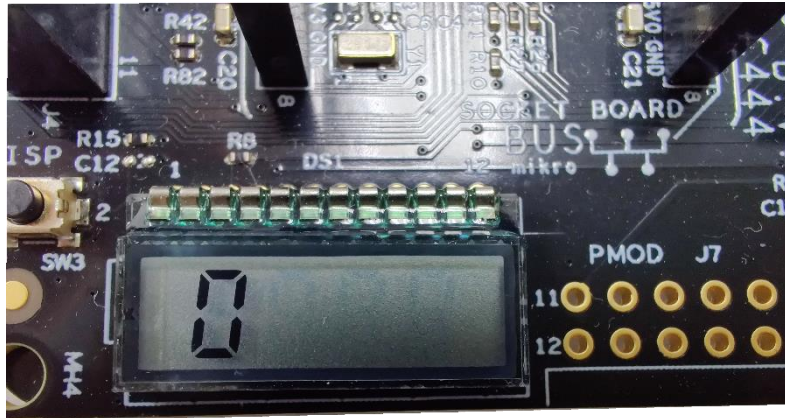


图7. 运行模式

5. UART终端也会显示功耗模式选择菜单，如图8所示。

```
COM22:9600baud - Tera Term VT
File Edit Setup Control Window Help

power mode switch example.

##### Power Mode Switch Demo #####

Core Clock = 48000000Hz
Power mode: RUN

Select the desired operation

Press 0 for enter: RUN      - Normal RUN mode
Press 1 for enter: WAIT    - Wait mode
Press 2 for enter: STOP    - Stop mode
Press 3 for enter: VLPR   - Very Low Power Run mode
Press 4 for enter: VLPW   - Very Low Power Wait mode
Press 5 for enter: VLPS   - Very Low Power Stop mode
Press 6 for enter: LLS/LLS3 - Low Leakage Stop mode
Press 7 for enter: VLLS0  - Very Low Leakage Stop 0 mode
Press 8 for enter: VLLS1  - Very Low Leakage Stop 1 mode
Press 9 for enter: VLLS3  - Very Low Leakage Stop 3 mode

Waiting for power mode select..
```

图8. UART终端上的功耗模式选择

6. 通过在UART终端中输入“0”到“9”来选择目标功耗模式。大多数功耗模式支持使用LPTMR或SW2按钮事件唤醒。在选择所需的唤醒源后，目标功耗模式被激活。图9所示为SLCD显示的当前功耗模式。

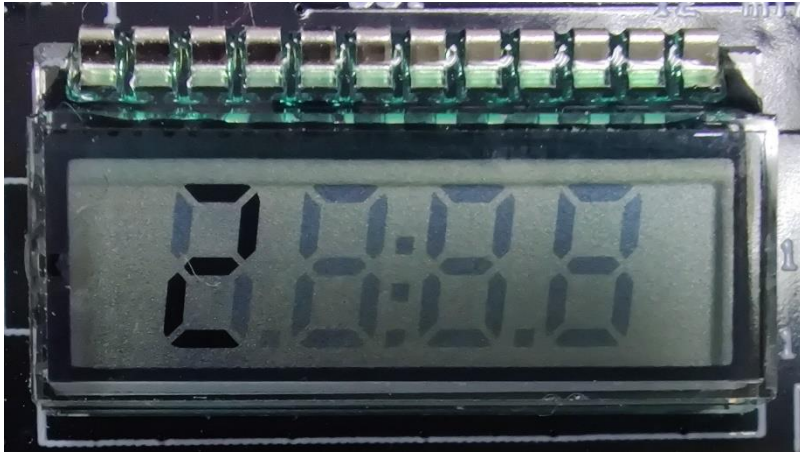


图9. 停止模式

7. 要唤醒MCU，请按**SW2**或等待LPTMR到期。然后，SLCD上的数字将变为“0”或“3”，具体取决于之前的功耗模式，如**图10**所示。

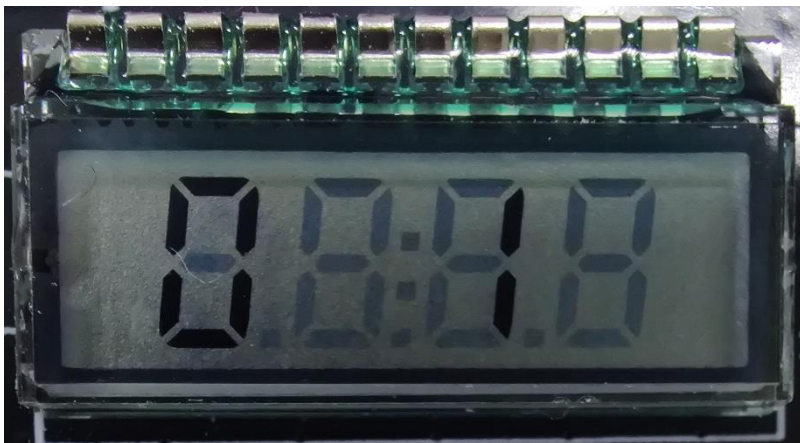


图10. 带软件令牌的运行模式

图10蕴含了以下内容：

- 左侧的“0”表示MCU返回到运行模式。当MCU从VLPS/VLPW唤醒到VLPR时，该值为“3”。
- 右侧的“1”表示保留了SRAM的内容。当SRAM掉电且内容丢失时，该值为“0”。
- 在VLLS0模式下，SLCD外设断电，因此在此模式下无LCD显示。LCD在唤醒后恢复工作。

8. MCU返回到运行模式或VLPR模式。

9. 在VLLSx模式下，MCU从复位处理程序中唤醒。

10. 要检查在这些模式下是否保留了SRAM的内容，请检查软件令牌：

- 如果保留了SRAM的内容（内存令牌有效），则当前电源模式旁会显示数字“1”。
- 如果SRAM的内容丢失，则显示数字“0”。

在切换功耗模式的操作过程中，用户可以读取万用表上的电流值，以显示实时功耗。

4 结论

当运行此应用程序演示时，测量了FRDM-MCXC444开发板上MCXC444的功耗。[表4](#)列出了所有测量值。

注：即使在低功耗模式下，[配OSC32时钟源的LLWU、SLCD](#)仍然处于工作状态，因为它们是为低功耗应用而设计的。

表4. MCXC444在各种功耗模式下的功耗

功耗模式	VDD_I	内存保留	注释
运行	8.36 mA	是	<ul style="list-style-type: none"> • 48MHz内核时钟 • UART启用
等待	3.26 mA	是	<ul style="list-style-type: none"> • 内核处于睡眠模式 • UART禁用 • NVIC唤醒
停止	0.16 mA	是	<ul style="list-style-type: none"> • 内核处于深度睡眠模式 • UART禁用 • AWIC唤醒
VLPR	0.91 mA	是	<ul style="list-style-type: none"> • 2MHz内核时钟 • UART启用
VLPW	0.09 mA	是	<ul style="list-style-type: none"> • 内核处于睡眠模式 • UART禁用 • NVIC唤醒
VLPS	3.9 μ A	是	<ul style="list-style-type: none"> • 内核处于深度睡眠模式 • UART禁用 • AWIC唤醒
LLS	3.4 μ A	是	<ul style="list-style-type: none"> • 内核处于深度睡眠模式 • UART禁用 • LLWU唤醒 • SLCD和OSC32启用
VLLS0	0.5 μ A	否	<ul style="list-style-type: none"> • 内核处于深度睡眠模式 • UART禁用 • LLWU唤醒 • SLCD和OSC32禁用
VLLS1	1.9 μ A	否	<ul style="list-style-type: none"> • 内核处于深度睡眠模式 • UART禁用 • LLWU唤醒 • SLCD和OSC32启用
VLLS3	2.9 μ A	是	<ul style="list-style-type: none"> • 内核处于深度睡眠模式 • UART禁用 • LLWU唤醒 • SLCD和OSC32启用

注：开发板功耗结果仅供参考。有关MCU功耗数据，请参阅其数据手册。

5 关于本文中源代码的说明

本文中所示的示例代码具有以下版权和BSD-3-Clause许可：

2024年恩智浦版权所有；在满足以下条件的情况下，可以源代码和二进制文件的形式重新分发和使用本源代码（无论是否经过修改）：

- 重新分发源代码必须保留上述版权声明、这些条件和以下免责声明。
- 以二进制文件形式重新分发时，必须在文档和/或随分发提供的其他材料中复制上述版权声明、这些条件和以下免责声明。
- 未经事先书面许可，不得使用版权所有者的姓名或参与者的姓名为本软件的衍生产品进行背书或推广。

本软件由版权所有者和参与者“按原样”提供，不承担任何明示或暗示的担保责任，包括但不限于对适销性和特定用途适用性的暗示保证。在任何情况下，无论因何种原因或根据何种法律条例，版权所有或参与者均不对因使用本软件而导致的任何直接、间接、偶然、特殊、惩戒性或后果性损害（包括但不限于采购替代商品或服务；使用损失、数据损失或利润损失或业务中断）承担责任，无论是因合同、严格责任还是侵权行为（包括疏忽或其他原因）造成的，即使事先被告知有此类损害的可能性也不例外。

6 修订历史

[表5](#)汇总了本文的修订情况。

表5. 修订历史

文档ID	发布日期	说明
AN14332 v.1.0	2024年7月9日	首次公开发布

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

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.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at <https://www.nxp.com.cn/profile/terms>, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. 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 B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

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 and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) 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.

目录

1	介绍	2
2	MCXC444 MCU的功耗模式	2
2.1	Arm Cortex-M0+核的基本功耗模式	2
2.2	MCXC444中的扩展功耗模式	3
3	测量各种功耗模式下的电流	7
3.1	开发板设置	7
3.2	软件设计	8
3.2.1	通过软件切换功耗模式.....	8
3.2.2	在低功耗模式下保留SRAM的内容.....	10
3.3	运行应用程序演示工程.....	11
4	结论	14
5	关于本文中源代码的说明	15
6	修订历史	15
	法律声明	16

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.