

AN11027

LPCXpresso 环境下的 LPC1100 低功耗模式和唤醒时钟

Rev. 1-7/2/2011 应用手册

文档信息

信息	内容
关键字	LPC11XX,LPC11XX(L),LPC11CXX,低功耗模式,功耗,唤醒,代码示例,LPCXpresso
摘要	该应用手册介绍了 LPC11xx 系列 ,LPC11xx(L) 系列 和 LPC11Cxx 系列的各种低功耗模式。进入低功耗模式、唤醒状态，可以有效降低功耗。本文档同时提供了一个软件示例，使用 LPC1114 LPCXpresso 板进入低功耗模式测量功耗和唤醒时钟。

深圳市伟博创科技有限公司

Tel:(0755)87240703 Fax:83240724

<http://www.weboch.com.cn>

<http://www.longdouble.com>

1. 简介

基于 ARM Cortex-M0 CPU 架构, LPC1100 微控制器系列 (8/16 位应用), 高性能、低功耗、简易指令、存储器寻址, 与现有的 8/16 架构相比, 代码体积更小。

LPC1100 系列外设包括 32KB flash 存储器, 8KB 数据存储器, 一个 C-CAN 控制器 (LPC11Cxx), 一个高速+IIC 总线接口, 一个 RS-485/EI-485UART, 两个 SPI 接口 (SSP 特性), 4 个通用定时器, 一个 10 位 ADC, 以及 42 个通用 I/O 引脚。

LPC11Cxx 系列包含片上 C-CAN 驱动和通过 C-CAN 在线系统编程 flash 工具。LPC11C2x 系列有一个片上高速 CAN 收发器。

LPC11xx (L) 系列包含片上功率配置部件。

LPC1100 系列定位于一系列广泛的应用, 包括 eMetering、照明、工业网络、报警系统以及白色家电。

本文档介绍了 LPC11xx 系列, LPC11xx(L) 系列和 LPC11Cxx 系列的各种低功耗模式。进入低功耗模式、唤醒状态, 可以有效降低功耗。本文档同时提供了一个软件示例, 使用 LPC1114 LPCXpresso 板进入低功耗模式测量功耗和唤醒时钟。

本文档包括以下要点:

- (1) 低功耗模式
- (2) 进入低功耗模式
- (3) 唤醒状态
- (4) 降低功耗
- (5) LPC1114 LPCXpresso 低功耗模式演示

2. 低功耗模式

LPC11xx、LPC11xx (L)、LPC11Cxx 系列有三种省电模式: 睡眠模式、深度睡眠模式和深度掉电 (Power-down) 模式。以下内容包含了低功耗模式的特性和配置。

2.1.1 睡眠模式

在睡眠模式, ARM Cortex-M0 内核系统时钟停止, 指令的执行暂停, 直到复位或者发生中断。

外设功能, 如果选择 SYSAHBCLKCTRL 寄存器计时, 在睡眠模式继续操作, 则会产生中断, 恢复处理器运行。睡眠模式取消了动态功率 (用于处理器自身、存储器系统和相关的控制器、内部总线)。处理器状态和寄存器、外设寄存器、内部 SRAM 的值保持不变, 引脚的逻辑电平不变。

2.1.2 深度睡眠模式

在深度睡眠模式, 与睡眠模式一样, 处理器系统时钟不可用。所有的模拟块都掉电, 除了 BOD 电路和看门狗震荡器 (必须在深度睡眠模式期间, 在 PDSLEEPCFG 寄存器中选定或取消选定)。详细内容请参考 3.5 部分。

深度睡眠模式取消了所有用于 flash、模拟外设的电源, 用于处理器自身、存储器系统

和相关控制器、内部总线的动态电源。处理器状态和寄存器、外设寄存器以及内部 SRAM 的值保持不变，引脚的逻辑电平不变。

2. 1. 3 深度掉电模式

在深度掉电模式，WAKEUP 引脚的异常切断了整个芯片的电源和时钟。在深度掉电模式期间，不保留 SRAM 和寄存器的内容，除了少数保存在电源管理单元中的 5 个 32 位通用寄存器的数据。

在深度掉电模式下，所有的功能引脚都是三态的，除了 WAKEUP 引脚。

3. 进入低功耗模式

这部分描述了 LPC11xx、LPC11xx (L)、LPC11Cxx 系列进入三种低功耗模式的机制（睡眠模式、深度睡眠模式、深度掉电模式）。

3. 1 系统控制寄存器（SCR）

SCR 寄存器控制进入和退出低功耗模式的特性。

在 Cortex-M0 系统控制寄存器（SCR）中，通过设置 SLEEPDEEP 位，进入睡眠模式和深度睡眠模式。

位描述如图 1 所示。

Bits	Name	Function
[31:5]	-	Reserved.
[4]	SEVONPEND	Send Event on Pending bit: 0 = only enabled interrupts or events can wake-up the processor, disabled interrupts are excluded 1 = enabled events and all interrupts, including disabled interrupts, can wake-up the processor. When an event or interrupt enters pending state, the event signal wakes up the processor from WFE. If the processor is not waiting for an event, the event is registered and affects the next WFE. The processor also wakes up on execution of an SEV instruction.
[3]	-	Reserved.
[2]	SLEEPDEEP	Controls whether the processor uses sleep or deep sleep as its low power mode: 0 = sleep 1 = deep sleep.
[1]	SLEEPONEXIT	Indicates sleep-on-exit when returning from Handler mode to Thread mode: 0 = do not sleep when returning to Thread mode. 1 = enter sleep, or deep sleep, on return from an ISR to Thread mode. Setting this bit to 1 enables an interrupt driven application to avoid returning to an empty main application.
[0]	-	Reserved.

图 1 系统控制寄存器（SCR，0xE000ED10）

LPCXpresso 环境下的 LPC1100 低功耗模式和唤醒时钟

在 Cortex-M0 系统控制寄存器 (SCR) 中, 如果 SLEEPDEEP 位是 0, 则是睡眠模式。
在 Cortex-M0 系统控制寄存器 (SCR) 中, 如果 SLEEPDEEP 位是 1, 则是深度睡眠模式。

3. 2 电源控制寄存器 (PCON)

PCON 寄存器选择哪种睡眠模式 (睡眠模式或深度睡眠模式), 或者进入深度掉电模式。PCON 分别提供了睡眠/深度睡眠模式和深度掉电模式的标志位, 参看图 2。

Table 47. Power control register (PCON, address 0x4003 8000) bit description

Bit	Symbol	Value	Description	Reset value
0	-	-	Reserved. Do not write 1 to this bit.	0x0
1	DPDEN		Deep power-down mode enable	0
		0	ARM WFI will enter Sleep or Deep-sleep mode (clock to ARM Cortex-M0 core turned off).	
		1	ARM WFI will enter Deep-power down mode (ARM Cortex-M0 core powered-down).	
7:2	-	-	Reserved. Do not write ones to this bit.	0x0
8	SLEEPFLAG		Sleep mode flag	0
		0	Read: No power-down mode entered. LPC111x/LPC11C1x is in Active mode. Write: No effect.	
		1	Read: Sleep/Deep-sleep or Deep power-down mode entered. Write: Writing a 1 clears the SLEEPFLAG bit to 0.	
10:9	-	-	Reserved. Do not write ones to this bit.	0x0
11	DPDFLAG		Deep power-down flag	0x0
		0	Read: Deep power-down mode not entered. Write: No effect.	0x0
		1	Read: Deep power-down mode entered. Write: Clear the Deep power-down flag.	0x0
31:12	-	-	Reserved. Do not write ones to this bit.	0x0

图 2 电源模式控制寄存器

3. 3 中断等待 (WFI) 指令

设置完 SLEEPDEEP 位和 PCON 寄存器, 执行 WFI 指令将直接进入低功耗省电模式。WFI 指令是 Cortex-M0 指令, 不能用 ANSIC 直接访问。CMSIS (Cortex 微控制器软件接口标准) 提供了一个内置函数, 生成 WFI 指令, 支持 C 编译器。如果 C 编译器不支持 WFI 内置函数, 用户必须使用汇编代码执行 WFI 指令。

3. 4 进入睡眠模式的编程步骤

以下是进入睡眠模式的步骤:

- (1) PCON 寄存器的 DPDEN 位必须设置为 0。
- (2) ARM Cortex-M0 SCR 寄存器的 SLEEPDEEP 位必须设置为 0。
- (3) 使用 ARM Cortex-M0 中断等待指令 (WFI)。

图 3 为进入睡眠模式的示例代码

```

/* Clear the Deep Power down flag from the PMU */
LPC_PMU->PCON |= (1<<11);

/* Specify Sleep mode before entering mode */
SCB->SCR    &= ~(1<<2);    //Clear SLEEPDEEP bit

/* Enter Sleep mode */
__WFI();

```

图 3 示例代码（睡眠模式）

3. 5 进入深度睡眠模式的编程步骤

以下为进入深度睡眠模式的步骤：

(1) PCON 寄存器的 DPDEN 位必须设置为 0。

(2) 在 PDSLEEPCFG 寄存器深度睡眠模式下选择电源配置。

当器件进入深度睡眠模式，这个寄存器控制看门狗震荡器和 BOD 电路。在进入深度睡眠模式之前，必须至少一次使用下图中的值初始化这个寄存器。初始化失败，可能导致微控制器未定义错误。图 4 所列的是 PDSLEEPCFG 寄存器所允许的值。

Allowed values for PDSLEEPCFG register		
Configuration	WD oscillator on	WD oscillator off
BOD on	PDSLEEPCFG = 0x0000 18B7	PDSLEEPCFG = 0x0000 18F7
BOD off	PDSLEEPCFG = 0x0000 18BF	PDSLEEPCFG = 0x0000 18FF

图 4 PDSLEEPCFG 寄存器设置（深度睡眠模式）

- a. 如果定时器需要控制自身唤醒，确保在 PDRUNCFG 寄存器中设置看门狗震荡器，并在 MAINCLKSEL 寄存器中选择看门狗震荡器的时钟源（细节请参考 LPC11XX 用户手册）。
- b. 看门狗震荡器可以运行在深度睡眠模式，为看门狗定时器或者通用定时器提供时钟，如果它们需要自唤醒。更多细节请参考 3.7 部分。在这种情况下，看门狗震荡器模拟输出频率必须设置为它的最低值（WDTOSCCTRL 中位 FREQSEL=0001，参考 LPC11XX 用户手册），进入深度睡眠模式之前，在 SYSAHBCLKCTRL 寄存器中设置外设时钟为不可用（除了定时器时钟）。如果在深度睡眠模式下运行看门狗震荡器，将产生额外的电流损耗。
- c. 如果没有定时器自唤醒需要，且看门狗震荡器关闭，确保在 PDRUNCFG 寄存器中设置对 IRC 的供电，并在进入深度睡眠模式前在 MAINCLKSEL 寄存器中为 IRC 选择时钟源。这样能确保顺利关闭系统时钟（更多细节请参考 LPC11XX 用户手册）。
- d. 在深度睡眠模式下，使能 BOD 电路可以在低压下保护器件。虽然，在深度睡眠模式下，BOD 电路会产生额外电流损耗。

LPCXpresso 环境下的 LPC1100 低功耗模式和唤醒时钟

- (3) 唤醒之后，在 PDAWAKECFG 寄存器中选择电源配置。可以在寄存器中对该位进行编程，以决定芯片从深度睡眠模式唤醒时进入的状态。
- (4) 如果使用外部引脚作为唤醒引脚，则在开始逻辑寄存器中使能并清除该唤醒引脚，并在 NVIC 中使能匹配开始逻辑的中断。
- (5) 在 SYSAHBCTRL 寄存器中，禁用所有的外设，除了计数器/定时器或 WDT（如果需要）。
- (6) 在 ARM Cortex-M0 SCR 寄存器中，SLEEPDEEP 位写入 1。
- (7) 使用 ARM Cortex-M0 中断等待指令（WFI）。

图 5 为进入深度睡眠模式的示例代码

```

/* Turn on all the IRC & Flash */
LPC_SYSCON->PDRUNCFG &= ~(1<<0 | 1<<1 | 1<<2);

/* Switch MAINCLKSEL to IRC */
LPC_SYSCON->MAINCLKSEL = 0;
LPC_SYSCON->MAINCLKUEN = 0;
LPC_SYSCON->MAINCLKUEN = 1;
while (!(LPC_SYSCON->MAINCLKUEN & 0x01));

/* Ensure DPDEN is disabled in the power control register */
LPC_PMU->PCON = (1<<11); //Clear DPDFLAG if it was set

/* Clear the Deep Sleep Flag */
LPC_PMU->PCON |= (1<<8);

/* All OFF */
LPC_SYSCON->PDSLEEPFCFG |= 0x000018FF;

/* Specify peripherals to be powered up again when returning from deep sleep mode */
LPC_SYSCON->PDAWAKECFG = LPC_SYSCON->PDRUNCFG;

/* Specify the start logic to allow the chip to be waken up */
LPC_SYSCON->STARTAPRPO &= ~(1<<(2)); // Falling edge
LPC_SYSCON->STARTRSRPOCLR |= (1<<(2)); // Clear pending bit
LPC_SYSCON->STARTERPO |= (1<<(2)); // Enable Start Logic

NVIC_ClearPendingIRQ(WAKEUP2_IRQn);
NVIC_EnableIRQ(WAKEUP2_IRQn);

/* Specify Deep Sleep mode before entering mode */
SCB->SCR |= (1<<2); //Set SLEEPDEEP bit

/* Enter Deep Sleep mode */
__WFI();

```

图 5 代码示例（深度睡眠模式）

3. 6 进入深度掉电模式的编程步骤

进入深度掉电模式的步骤如下：

- (1) 在 PCON 寄存器中，DPDEN 位写入 1。
- (2) 保存通用寄存器中的数据（更多细节请参考 LPC11XX 用户手册）。
- (3) ARM Cortex-M0 SCR 寄存器，SLEEPDEEP 位写入 1。
- (4) 在进入深度掉电模式之前，在 PDRUNCFG 寄存器中，IRCOUT-PD 和 IRC-PD 位写入 0，以确保对 IRC 供电。
- (5) 使用 ARM Cortex-M0 中断等待指令（WFI）。

请注意，在进入深度掉电模式之前，WAKEUP 引脚需要外部拉高。

```
/* Specify DPDEN to power control register */
LPC_PMU->PCON      = (1<<1) | (1<<11);

/* Specify Deep Power-down mode before entering mode */
SCB->SCR           |=      (1<<2);      //Set SLEEPDEEP bit

/* Enable IRC before deep power-down mode */
LPC_SYSCON->PDRUNCFG &= ~( (1<<0) | (1<<1));

/* Enter Deep Power-down mode */
__WFI();
```

图 3 示例代码（深度掉电模式）

3. 7 唤醒

以下部分描述了 LPC11xx、LPC11xx (L)、LPC11Cxx 系列每种模式的唤醒。

3. 7. 1. 1 从睡眠模式唤醒

当 NVIC 中断或者复位发生时，自动退出睡眠模式。关于哪个中断源与 NVIC 连接，请参考 LPC11XX 用户手册。中断唤醒后，微控制器返回到最开始的电源配置状态（由 PDRUNCFG 和 SYSAHBCLKDIV 寄存器决定）。如果发生复位，微控制器进入激活模式的默认状态。

3. 7. 1. 2 从深度睡眠模式唤醒

微控制器从深度睡眠模式唤醒有以下几种方式：

- (1) 连接到开始逻辑（start logic）的外部引脚信号。某一端口引脚可以作为开始逻辑的输入。关于有哪些端口引脚连接到开始逻辑并可作为唤醒引脚，请参考 LPC11XX 用户手册。开始逻辑不需要任何时钟，如果在 NVIC 中使能从深度睡眠模式唤醒，则可以产生中断。
- (2) 开始逻辑的输入信号（由某一通用定时器外部输入引起的事件所产生）。引脚（保持定时器匹配功能）在 NVIC 中设置为开始逻辑的输入，在 SYSAHBCLKCTRL 寄存器中使能相应的定时器，且看门狗震撼器运行在深度睡眠模式（更多细节请参考 LPC11XX 用户手册）。当触发了相匹配的输入时，唤醒机制允许器件自唤醒。如图 7 所示，反馈到开始逻辑，引发器件自唤醒。这种方式不需要外部触发，减少了板上的外部元件。用这种方法所花的唤醒时间比第一种方法长。关于唤醒时间将在演示部分讨论。

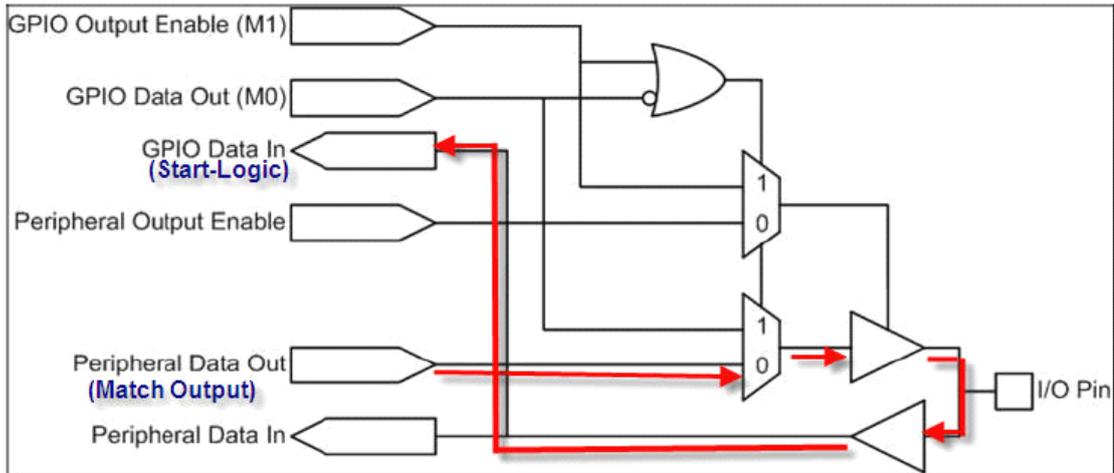


图 7 自唤醒框图

- (3) BOD 电路复位。此时，需要在 PDSLEEPCFG 寄存器中使能 BOD 电路，且在 BODCTRL 寄存器中使能 BOD 复位（更多细节请参考 LPC11XX 用户手册）。
- (4) 看门狗定时器复位。此时，看门狗振荡器必须运行在深度睡眠模式（参看 PDSLEEPCFG 寄存器），且在 SYSAHBCLKCTRL 寄存器中使能看门狗定时器外设。
- (5) 外部 RESET 引脚。

3. 7. 1. 3 从深度掉电模式唤醒

拉低 WAKEUP 引脚从深度掉电唤醒，芯片复位。WAKEUP 引脚最小脉冲宽度为 50ns。

- (1) 当 WAKEUP 引脚从高电平到低电平转换时，唤醒有效（电平触发）。
 - a. 电源管理单元（PMU）打开片上电压调节器。当内核电压达到电源复位（power-on-reset）跳变点，系统复位，芯片重新引导。
 - b. 所有寄存器（除了 GPREG0-GPREG4、PCON）回到复位状态。
- (2) 一旦芯片引导完成，可以从 PCON 寄存器中读取深度掉电标志位，以确认由唤醒事件引发的复位。
- (3) 清除 PCON 寄存器深度掉电标志位。
- (4) （可选）读取通用寄存器保存的数据。

请注意，RESET 引脚在深度掉电模式下没有任何功能。

3. 8 降低功耗补充材料

进一步降低电流损耗，可以考虑以下几点：

3. 8. 1. 1 CPU 时钟频率

可以根据需要控制 CPU 时钟频率。

- (1) 改变时钟源：内部振荡器（12MHz +/-1%）、系统振荡器（1MHz to 25MHz），看门狗振荡器（7.8kHz to 1.7MHz +/-40%）。
- (2) 重设 PLL 的值，和/或变更 AHB Clock Divider（SYSAHBCLKDIV）的值。

LPCXpresso 环境下的 LPC1100 低功耗模式和唤醒时钟

- (3) 降低 PLL 输出频率 (FCCO, 156MHz to 320MHz) 可以降低功耗。
- (4) 8 位系统 AHB Clock divider 寄存器 (SYSAHBCLKDIV) 可以降低 CPU 的操作到某一低的频率, 临时降低功耗而无需关闭 SYS PLL。见图 8 所示。

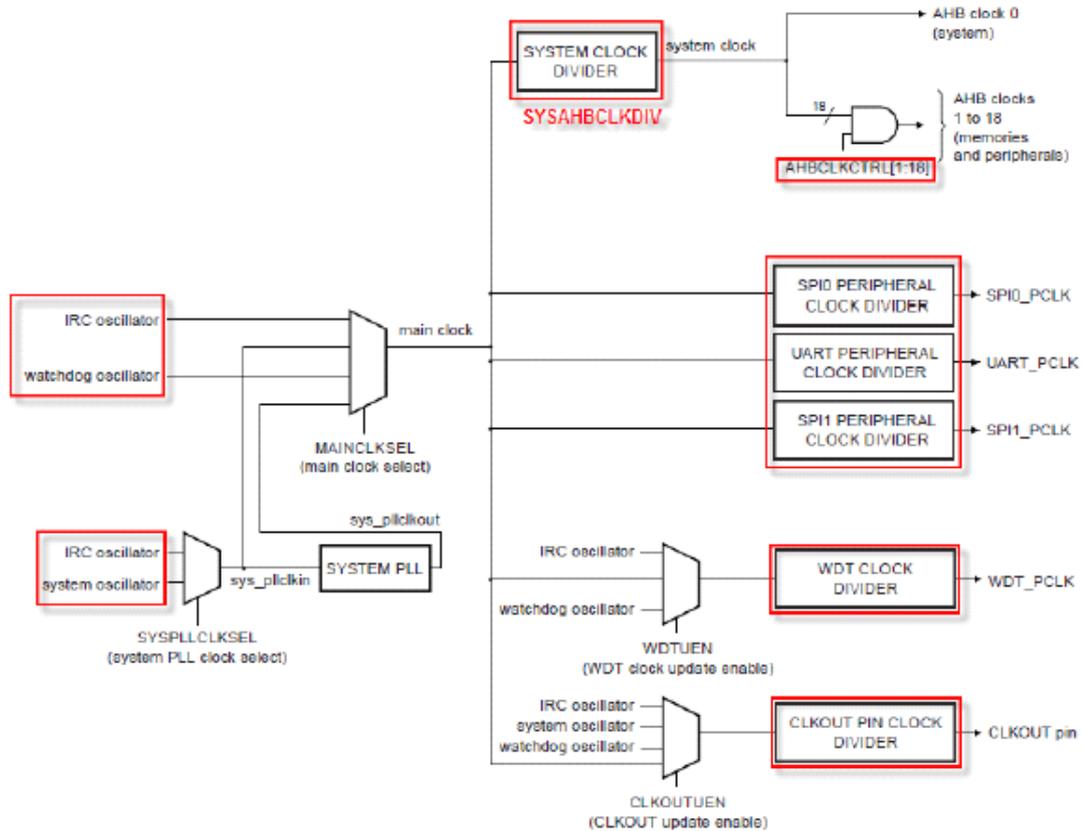


图 8 时钟发生单元 (CGU) 框图

3. 9 系统 AHB 时钟控制寄存器 (AHBCLKCTRL)

如图 8 所示, SYSAHBCLKCTRL 寄存器使能系统和外充模块的时钟。根据需要, 用户可以使用该寄存器禁用未使用的外设时钟, 以达到降低功耗的目的。更多细节请参考 LPC11XX 用户手册。

3. 10 外设时钟除法器

选定的外设有各自的时钟除法器 (SPI0CLKDIV, UARTCLKDIV, SPI1CLKDIV, WDTCLKDIV, CLKOUTDIV), 可以控制提供给相应外设的时钟信号的频率 (见图 8)。根据实际需要, 用户可以使用除法器降低外设时钟频率或者关闭时钟以降低功耗。更多细节请参考 LPC11XX 用户手册。

3. 11 掉电配置寄存器 (PDRUNCFG)

PDRUNCFG 寄存器包含模拟块电源开关的控制位, 通过模拟外设可以消除电源消耗。更多细节请参考 LPC11XX 用户手册。

3. 12 其他

3. 12. 1 软件

大部分嵌入式应用都以 while(1) loop 结束, 用于中断。此时, 仍然从片上 flash 获取代

码并执行，增加了功耗。一种较好的解决方案是选择睡眠省电模式，并等待中断。外设的中断可以把器件从睡眠模式唤醒。内核在等待中断时保持睡眠模式，可以有效的降低功耗。

3. 12. 2 端口引脚

可以增加一些步骤，在低功耗模式下降低电流损耗。LPC11xx、LPC11xx (L)、LPC11Cxx 系列的通用端口引脚，默认可编程内部拉高。在进入低功耗模式之前，可以通过以下几种方法降低 IO 的电流：

- (1) 所有通用 I/O 引脚 (GPIO) 默认带有输入上拉电阻。首先，使用 IOCON 寄存器，禁用所有通用端口引脚内部上拉 (pull-ups)。其次，设置 I/O 为 GPIO 输出，低电平。确保引脚没有外部拉高。
- (2) 所有通用 I/O 引脚 (GPIO) 默认带有输入上拉电阻。首先，使用 IOCON 寄存器，禁用所有通用端口引脚内部上拉 (pull-ups)。其次，设置 I/O 为 GPIO 输出，高电平。确保引脚没有外部拉低。
- (3) 使用 IOCON 寄存器，首先，禁用所有通用端口引脚内部上拉 (pull-ups)。其次，设置 I/O 为 GPIO 输入且使用外部电阻，把 I/O 拉高或拉低。

请注意，在深度睡眠模式，端口引脚的状态不影响电流的损耗，以上提到的方法在这个模式下无需考虑。

3. 12. 3 CAN 引脚

LPC11C1x 器件，如果不使用 CAN 部件，CAN 引脚 (CAN-RXD 和 CAN-TXD) 建议外部拉低，以降低 IO 电流。

3. 12. 4 RESET-N/PIO0-0 引脚

LPC11xx、LPC11xx (L)、LPC11Cxx，在深度掉电模式时，RESET/PIO0-0 引脚需要通过外部 10K-47K 电阻拉高。如果引脚悬空，用户可以看到电流损耗的增加。

3. 12. 5 调试

用户应该知道的关于一些调度的限制。最重要的是，由于 Cortex-M0 集成方面的一些限制，LPC11xx、LPC11xx (L)、LPC11Cxx 在一般方式下，不能从睡眠、深度睡眠和深度掉电模式下唤醒。建议不要在这些模式下调试。一旦通过 SWD 接口下载应用程序，USB 和 SWD 之间的适配器要从板上移除。另一个问题是，调式模式改变了 Cortex-M0 CPU 处理降低功耗模式的方式。这将导致器件级的电源模式不同于通常模式下的操作。这些不同，意味着在调试时不能测试电源；在应用中，测试结果将比通常操作时要高。

4. 低功耗模式演示

4. 1 目标

本文档提供了在 LPCXpresso 平台下的两个低功耗模式的示例。

演示 1: UART 接口的低功耗模式 (睡眠、深度睡眠、深度掉电)

这个示例，允许用户进入低功耗模式且通过外部引脚唤醒。

演示 2: 自唤醒

这个示例在运行模式和深度睡眠模式下，配置了最低看门狗震荡器频率给 16 位定时器 0。器件从深度睡眠模式下重自唤醒，且在运行模式时触发一个端口引脚。两个示例都向用

LPCXpresso 环境下的 LPC1100 低功耗模式和唤醒时钟

户演示了如何测试电流和唤醒定时器。以下部分提供了在 LPCXpresso 平台下建立低功耗模式例程的必要步骤。这两个工程都符合 CMSIS (Cortex 微控制器软件接口标准)。

4. 2 LPCXpresso 示例

4. 2. 1. 1 配置

- (1) LPCXpresso IDE
- (2) LPCXpresso 目标板 (LPC1114) (参看图 9)
- (3) LPCXpresso 底板 (Base Board) (参看图 11)

LPCXpresso 底板 (Base Board) 包含一个 USB-Serial 桥芯片 (FTDI FT232R), 把 LPCXpresso 目标板 LPC1114 的 UART 通道连接到 PC 上的虚拟 COM (通过 USB)。

注: 只有演示 1 需要底板, 使用 UART 接口显示电源模式选择菜单。

4. 2. 1. 2 终端窗口 (Terminal window)

在演示 1 中, 需要像 TeraTerm 或者 HyperTerminal 这样的终端窗口来显示进入低功耗模式的选项。

4. 2. 1. 3 LPCXpresso 目标板 (LPC1114)

电流测试

两个演示都需要 LPCXpresso 目标板做以下的步骤:

- (1) LPCXpresso 目标板包含一个集成 JTAG 调试器 (LPC-Link)。缺省情况下, LPC1114 和 LPC-Link 之间共享电源。切断 J4.1 和 J4.2 之间的跟踪连接, 以分开电源; 然后, 用户可以在 J4.2 连接一个外部电源, 以用电流计测量电流。在向 LPC1114 提供电源的 VIO-3V3X 引脚上测量电流。参看图 9。
- (2) 因为示例代码把所有的引脚配置为 GPIO 输出, 低电平; 所以需要移除 RESET/PIO0-0 引脚外部 12K 上拉电阻。参看图 9。

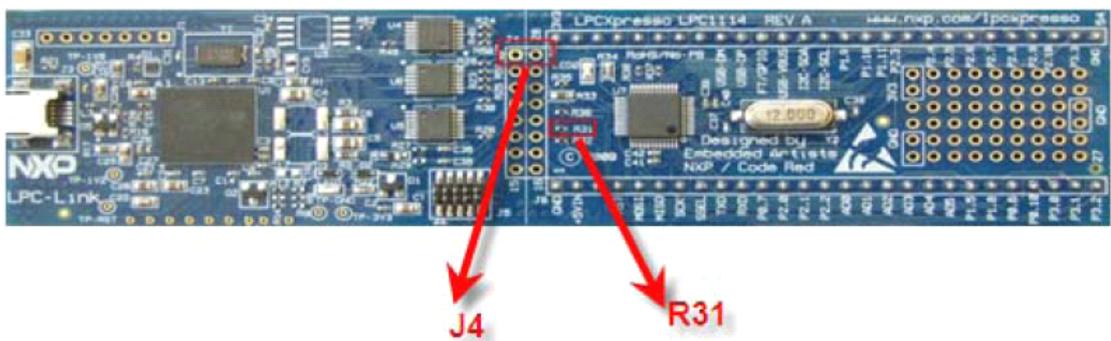


图 9 LPCXpresso LPC1114 目标板

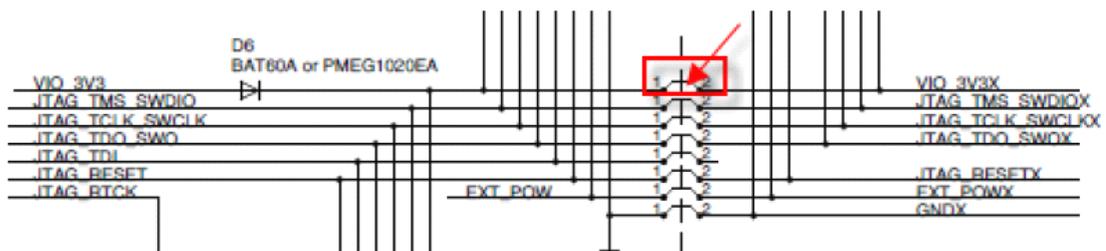


图 10 LPCXpresso 目标板原理图

4. 2. 1. 4 LPCXpresso 底板

如前所述，只有演示 1 需要底板，建立步骤如下：

运行应用程序

- (1) 移除 PIO0-2 和 BL-EN 之间的跳线
- (2) 连接 PIO0-2 和 BL-EN 之间的跳线
PIO0-2 作为睡眠和深度睡眠模式的唤醒源
‘BL-EN’ (SW3) 按钮用于睡眠和深度睡眠模式
- (3) 移除 PIO1-4 的跳线
PIO1-4 复用唤醒功能
‘WAKEUP’ (SW2) 按钮用于从深度掉电模式唤醒
- (4) 保持 J7 的 3 个跳线的连接（默认）。
这些跳线导通 UART 引脚和 FTDI 芯片。
- (5) 移除 J54 上的两个跳线
这样就移除了从 USB 到 UART 桥的 DTR 和 RTS 信号的控制。

测量电流

- (6) 完成了以上 4 步，再移除 LPCXpresso 底板上的剩余跳线就可以测量电流。
参看图 11 完整的跳线设置

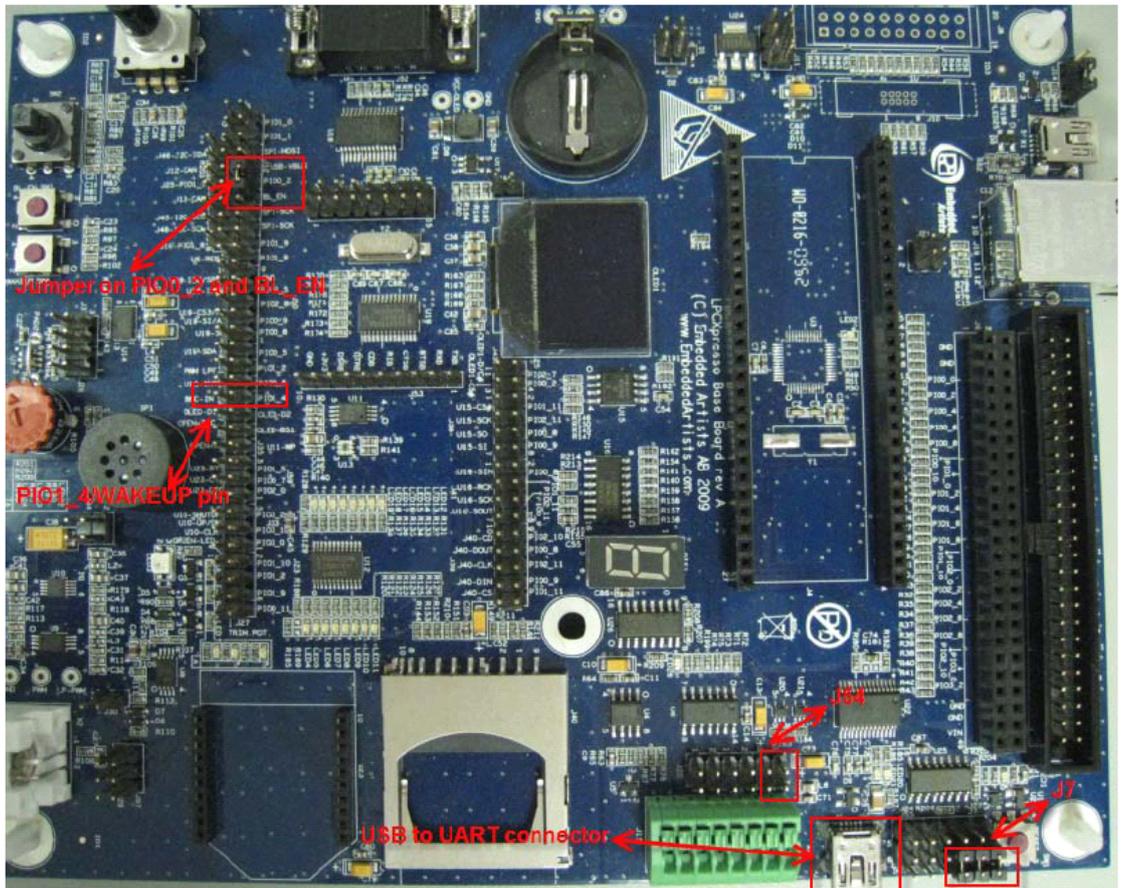


图 11 LPCXpresso 底板

4. 2. 1. 5 LPCXpresso 演示 1 的应用建立

如上所述，演示 1 允许用户通过 UART 接口进入低功耗模式，通过外部端口引脚实现唤醒。

建立步骤如下：

- (1) 连接 USB 线，给 LPCXpresso LPC1114 目标板上电
- (2) 打开 LPCXpressoIDE，引入 LPC1100wakeup.zip 文件。
- (3) 选择 “the Build'all projects'(debug)” 建立工程
- (4) 选择 “the Debug 'wake-up'(Debug)” 下载代码
- (5) 终止调试部分，一旦代码下载完成
- (6) 从目标板上移除 USB 线
- (7) 把 LPCXpresso 目标板附加到 LPCXpresso 底板上
- (8) 连接 USB 线，给 LPCXpresso 目标板上电
- (9) 把 USB 线连接到 USB-UART 连接器（在 LPCXpresso 底板上）。参看图 11
- (10) 用户会被两次提示安装 FTDI 驱动，以实现 USB-Serial 芯片（FT232R）功能。
 - a. 选择以下路径
C:\nxp\lpcxpresso_3.3\Drivers\RedProbe
 - b. 选择以下文件：
C:\nxp\lpcxpresso_3.3\Drivers\RedProbe\i386\ftdibus.sys
C:\nxp\lpcxpresso_3.3\Drivers\RedProbe\i386\ftser2k.sys
 - c. 安装完成后可以在设备管理/属性看到以下内容（图 12）：

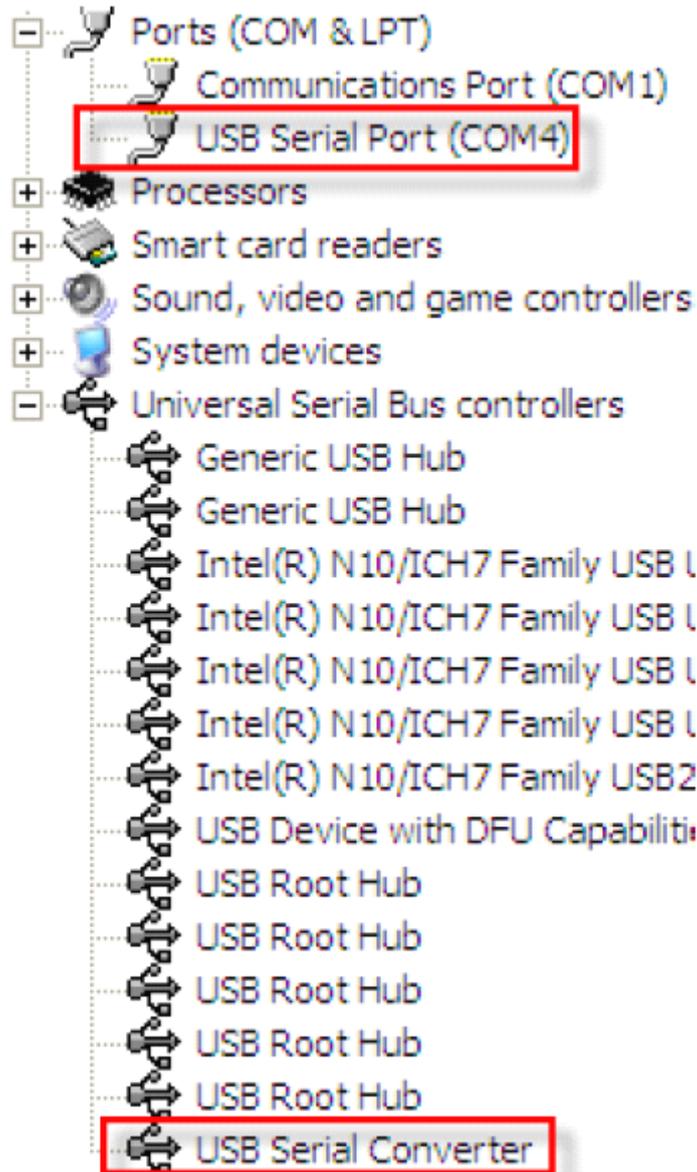


图 12

- (11) 一些端终程序需要使用较低的端口号，如 1-5 之间。很多情况下，USB 串口的 COM 端口号都比这高，因此，需要手动更改。如果是这种情况，按下列方法操作。更多细节可以参考 LPCXpresso 底板用户手册。
- A. 进入设备管理器、端口列表。
 - B. 右键单击 USB 串口，选择属性。
 - C. 进入端口设置(Port Settings)->Advanced to change COM Port Number(如下图所示)。端口号根据终端程序而变化。本例子中，端口号变为 COM4。

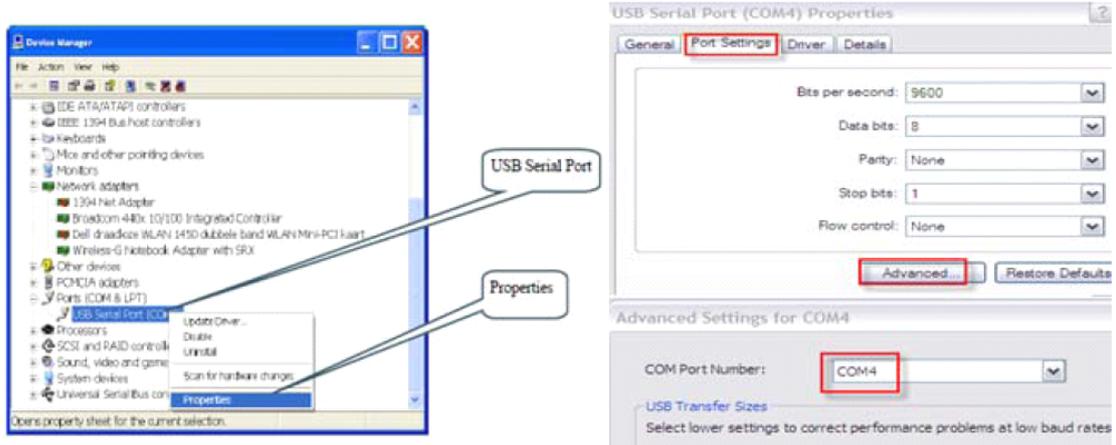


图 13 改变 USB 串口 COM 号

(12) 演示 1 使用终端程序以及串口设置，图 14

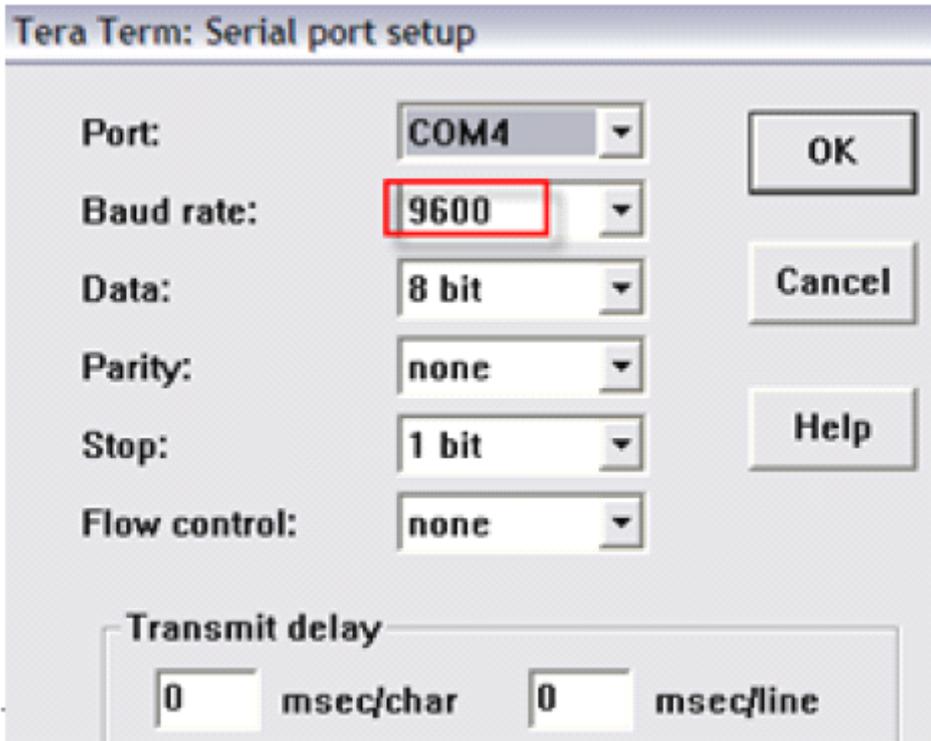


图 14

(13) 按下 RESET 按钮

(14) 代码输出见图 15。该菜单展示了如何进入 3 种低功耗模式

```

ENTER WHICH MODE YOU WANT TO ENTER - LPC1100
SLEEP mode
  'a' - IRC (12MHz)
  'b' - System Oscillator (12MHz)
  'c' - System Oscillator with PLL (50MHz)

DEEP SLEEP mode
  'd' - All OFF

DEEP POWER DOWN mode
  'e' - Enter DEEP POWER DOWN

Option:
    
```

图 15 低功耗模式

- (15) 唤醒器件
 - A. 按下 ‘BL-EN’ 按钮退出睡眠和深度睡眠模式。
 - B. 按下 WAKEUP 按钮退出深度掉电模式。
- (16) 在睡眠和深度睡眠模式，可以通过以下引脚测量唤醒时钟：
 - A. 触发引脚（PIO0-2）-让器件从睡眠或深度睡眠模式中退出。通过一个外部按钮触发该引脚（下降沿）来唤醒器件。
 - B. Strobe 引脚（PIO2-0-）-唤醒之后，器件返回运行模式，在运行 WAKEUP_IRQHANDLER()子程序期间，该引脚设为高电平。
- (17) 图 16 所示为触发引脚的下降沿（黄色波形）和 Strobe 引脚的上升沿（绿色波形）的唤醒时钟的区别。

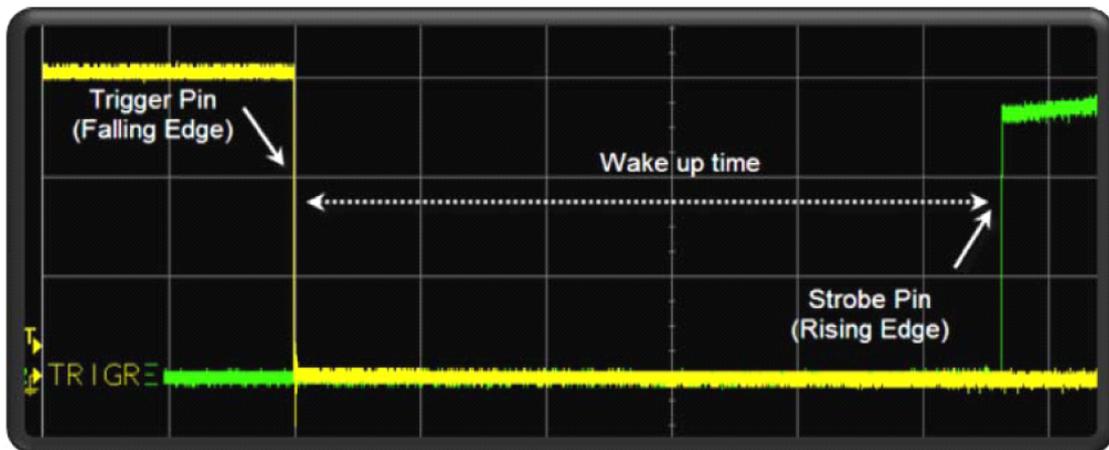


图 16 低功耗模式

4. 2. 1. 6 演示 1: 功耗测量和唤醒测量

表 1 所示为使用演示 1 所得的典型功耗和唤醒测量的结果。

Table 1. Typical power consumption (3.3 V, Temp = 25 °C)

Low power modes	Ivdd current	Wake-up
Sleep mode (IRC (12 MHz), all peripherals off)	2 mA	4.7 μ s
Sleep mode (System Osc (12 MHz), all peripherals off)	2 mA	4.7 μ s
Sleep mode (System Osc with PLL enabled (50 MHz), all peripherals off)	6 mA	1.18 μ s
Deep-sleep Mode (Watchdog Osc off and BOD off)	2 μ A (LPC11xx(L) series)	56 μ s
	6 μ A (LPC 11xx and LPC11Cxx series)	
Deep Power-down Mode (note 1)	220 nA	240 μ s

注意：从掉电模式唤醒类似于芯片的复位。该演示并不是为深度掉电模式的唤醒时间测量。使用演示 1，用户可以在复位处理函数中设置 `strobe` 引脚为高电平（参看 `the cr_startup_lpc11.c source` 文件）。在深度掉电模式中，唤醒时间为触发 `WAKEUP` 引脚（低电平）到 `strobe` 引脚进入高电平的这段时间。

4. 2. 1. 7 演示 2 应用的建立

如上所述，演示 2 在运行和深度睡眠模式下配置最低的看门狗震荡器频率给 16 位定时器 0。器件从深度睡眠模式自唤醒，并在运行模式下触发一个端口引脚。

建立步骤如下：

- (1) 连接 USB 线，给 LPCXpresso 目标板上电。
- (2) 打开 LPCXpressoIDE，引入 LPC1100 self wakeup.zip 文件。
- (3) 选择 the Build 'all projects' (Debug)建立工程。
- (4) 选择 the Debug 'wake-up'(Debug)下载代码。
- (5) 下载完毕则终止调试部分。
- (6) 给目标板重上电（power cycle）。
- (7) 使用下面的引脚可以测量唤醒时间：
 - A. 触发引脚（PIO0-8/CT16B0-MAT0）-定时器输出让器件退出深度睡眠模式。
 - B. Strobe 引脚（PIO0-7）-唤醒之后，器件返回运行模式，在 `WAKEUP_IRQHANDLER()`子程序运行期间，该引脚触发四次。
- (8) 图 17 所示波形为定时器和 GPIO 的输出。



1) Active Mode

$I_{vdd} = \sim 1.2 \text{ mA}$ when GPIO pin (strobe) goes high
 $I_{vdd} = \sim 429 \text{ uA}$ when GPIO pin (strobe) goes low

2) Deep Sleep (DS)

$I_{vdd} = 2 \text{ uA}$ (LPC11xx(L) series)
 $I_{vdd} = 6 \text{ uA}$ (LPC11xx and LPC11Cx series)

图 17 低功耗模式自唤醒演示

(8) 如图 18 所示, 触发引脚的上升沿和 strobe 引脚的第一个下降沿的唤醒时间的区别。

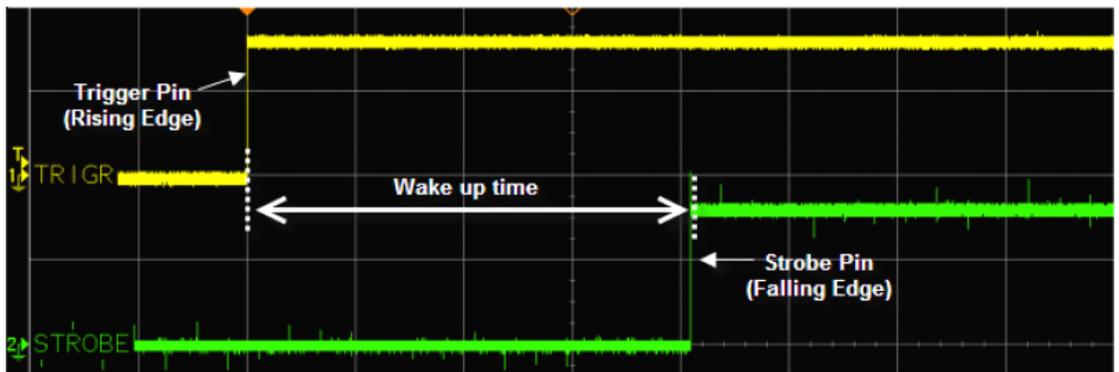


图 18 低功耗模式自唤醒演示

(9) 注意: 一旦演示 2 代码下载完毕, 器件进入和退出深度睡眠模式是同一个过程。用户必须首先在 ISP 模式下擦除或重编程 FLASH。关于如何进入 ISP 模式, 请参考 LPC11XX 用户手册。

4. 2. 1. 8 演示 2 的功耗测量和唤醒测量

表 2 为演示 2 的典型功耗和自唤醒测量的结果。

Table 2. Typical power consumption (3.3V, Temp = 25 °C)

Low power modes	Ivdd current	Wake-up
Deep-Sleep mode (Watchdog Oscillator (8.8 kHz) enabled, 16-bit Timer 0 on, BOD off)	3 μ A (LPC11xx(L) series) 7 μ A (LPC 11xx and LPC11Cxx series)	7.0 ms

5. 总结

LPC11xx、LPC11xx (L)、LPC11Cxx 系列提供了灵活的低功耗模式和唤醒功能。如表 1 所示，用户有多种低功耗选项，通过唤醒时间的取舍实现自己想要的功耗。用户可以根据自己的需要在功耗和唤醒速度做出相应的取舍。

另外，用户根据自唤醒的特性可以无需通过外部组件就能把器件从深度睡眠模式下唤醒。根据自唤醒特性，用户仍然可以降低功耗，但要以延长唤醒时间作为交换，如表 2 所示。为了实现从深度睡眠模式更快的唤醒，用户需要一个外部触发事件，在进入深度睡眠模式之前为 IRC 选择时钟源，并在深度睡眠模式下关闭看门狗震荡器。