AN11853

LPC1800 (with Flashless) Low Power Modes and Wake-up Times

Rev. 1.0 — 12 August 2016 **Application note**

Document information

Info	Content
Keywords	LPC1800, Flashless, Low Power Modes, Power Consumption, Wake-up, code example, LPCXpresso, MCB1800
Abstract	This application note introduces the various low power modes of the LPC18x0, the steps required to enter the low power modes, wake-up implementation, and helpful hints to reduce power consumption. This application note also provides a software example to enter the low power modes, and demonstrates how to measure the power consumption using the Keil MCB1800 board.

Revision history

Rev	Date	Description
1	20160812	Initial version.

Contact information

For additional information, please visit: http://www.nxp.com

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

1. Overview

The LPC1800 microcontroller family is based on the ARM Cortex-M3 CPU architecture for 32-bit microcontroller applications, offering performance and low power.
The LPC1800 operates at CPU frequencies up to 180 MHz. The peripheral complement of the LPC1800 family includes up to 200 kB of on-chip SRAM data memory (flashless parts) or up to 136 kB of on-chip SRAM and up to 1 MB of flash (parts with on-chip flash), 16 kB EEROM, a quad SPI Flash Interface (SPIFI), a State Configurable Timer (SCT) subsystem, two High-speed USB controllers, Ethernet, LCD, an external memory controller, one Fast-mode Plus I2C bus and one standard I2C-bus interface, four 550 UARTs with DMA support, two SSP controllers with DMA Support, four general purpose timers, one 10-bit ADCs, one 10-bit DAC both with DMA, and up to 164 general purpose I/O pins.
On-chip Power Management Controller (PMC) is included to control different power modes.
The LPC1800 family targets a wide range of applications, including eMetering, industrial, consumer, RFID readers, and white goods.

This application note describes the various low power modes of the LPC18x0 series, the steps required to enter the low power modes, wake-up procedure, and helpful hints to reduce power consumption of the device. Software examples included with this application note demonstrate how to enter the low power modes and how to measure the power consumption using a custom evaluation board.

The various topics covered in this application note are as follows:

1. LPC1800 Power Structure and Management.
2. Entering low power modes.
3. Wake-up implementation.
4. Additional hints to reduce power consumption.
5. Low power mode software example using the Keil MCB1800 board.

2. Power structure and management

2.1 Power domains

There are 7 power supplies on LPC18x0

1. $VDD_{(IO)}$ – power supply for IO pads.
2. $VDD_{(reg)(3V3)}$ – power supply for the core.
3. VDDA – power supply for analog peripherals.
4. VBAT – battery supply voltage.
5. USB0_VDDA3V3_DRIVER – separate analog power supply for USB0 driver.
6. USB0_VDDA3V3 – separate power supply voltage for USB0.

The LPC18x0 implements a separate VBAT power domain in order to allow turning off power to the bulk of the device while maintaining operation of the Real Time Clock. The VBAT pin supplies power only to the RTC domain. Whenever the device core power ($VDD_{(reg)}$) is present, that power is used to operate the RTC, causing no power drain from the battery when main power is available (see Fig 1).

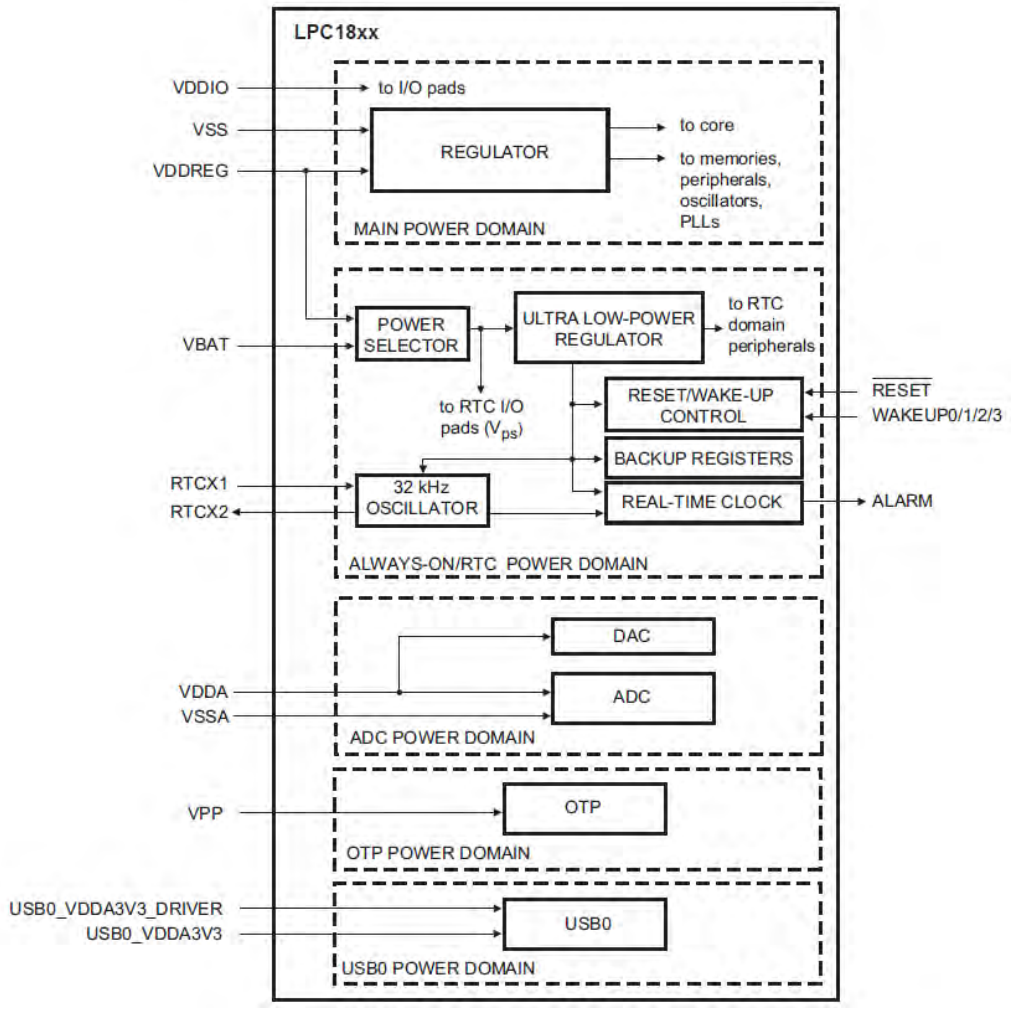

Fig 1. LPC1800 Power Structure

2.2 Power modes

On the LPC18x0 series, there are four reduced power down modes: Sleep, Deep-Sleep, Power-down, and Deep Power-down modes.

2.2.1 Sleep mode

In Sleep mode, the CPU clock is shut down to save power; however the peripherals can still remain active and fully functional. Sleep mode is entered by a WFI or WFE instruction if the SLEEPDEEP bit in the ARM Cortex-M3 system control register is set to 0.

The processor state and registers, peripheral registers, internal SRAM values, and the logic levels of the pins are maintained.

The part wakes up from sleep mode through any enabled interrupt in the NVIC when entered via WFI or any enabled event in the event router when enabled via WFE.

2.2.2 Deep-sleep mode

In Deep-sleep mode while the CPU clock and peripheral clocks are shut down to save power, logic states and SRAM memory are maintained. All analog blocks and the BOD control circuit are powered down.

Deep-sleep mode is entered by a WFI or WFE instruction if the SLEEPDEEP bit in the ARM Cortex-M3 system control register is set to 1 and the PD0_SLEEP0_MODE register is programmed with the value 0x0030 00AA.
When the LPC18x0 device wakes up from Deep-sleep mode, the 12 MHz IRC is used as the clock source for all base clocks. This requires that the clock source of all base clocks are switched to IRC and PLL's powered down before entering Deep-sleep mode.

The part can wake up from Deep-sleep mode only through a signal on any of the WAKEUP pins or a signal from the alarm timer or RTC. The wake-up signals are enabled in the event router.

2.2.3 Power-down mode

In Power-down mode the CPU clock and peripheral clocks are shut down, but logic states are maintained. All SRAM memory except for the upper 8 kB of the local SRAM located at 0x1008 0000, all analog blocks, and the BOD control circuit are powered down. The Power-down mode is entered by a WFI or WFE instruction if the SLEEPDEEP bit in the ARM Cortex-M3 system control register is set to 1 and the PD0_SLEEP0_MODE register is programmed with the value 0x0030 FCBA.

When the LPC18x0 device wakes up from Power-down mode, the 12 MHz IRC is used as the clock source for all base clocks. This requires that the clock source of all base clocks are switched to IRC and PLL's powered down before entering Power-down mode.

The part can wake up from Power-down mode only through a signal on any of the WAKEUP pins or a signal from the alarm timer or RTC. The wake-up signals are enabled in the event router.

2.2.4 Deep power-down mode

In Deep power-down mode the entire core logic is powered down and the logic state of the entire system including the I/O pads is lost. Only the logic in the RTC power domain remains active. The Deep power-down mode is entered by a WFI or WFE instruction if the SLEEPDEEP bit in the ARM Cortex-M3 system control register is set to 1 and the PD0_SLEEP0_MODE register is programmed with the value 0x0033 FF7F.

When the LPC18x0 wakes up from Deep power-down mode, the boot loader configures the PLL1 as the clock source running at 96 MHz and attempts to boot similar to that after a reset or power-up.

The part can wake up from Deep power-down mode only through a signal on any of the WAKEUP pins or a signal from the alarm timer or RTC. The wake-up signals are enabled in the event router.

2.2.5 Memory retention in Power-down modes

Fig 2 shows the memories that are preserved in the various low power modes:

Mode	64 kB + 32 kB local SRAM starting at 0x1000 0000	32 kB local SRAM starting at 0x1008 0000	upper 8 kB of local SRAM starting at 0x1008 8000	64 kB AHB SRAM starting at 0x2000 0000	256 byte backup registers at 0x4004 1000 (RTC power domain)
Sleep mode	yes	yes	yes	yes	yes
Deep-sleep mode	yes	yes	yes	yes	yes
Power-down mode	no	no	yes	no	yes
Deep power-down mode	no	no	no	no	yes

Fig 2. LPC1800 Memory Retention

All FIFO memory contained in the peripheral blocks (USB0/1, LCD, CAN0/1, Ethernet, USART0/2/3, and UART) is retained in Sleep mode and Deep-sleep mode but not in Power-down mode and Deep-power-down mode.

3. Entering low power modes and wake-up implementation

This section describes the mechanism to put the LPC18x0 into the four low power modes (sleep, deep-sleep, power-down, and deep power-down) and covers the available wake-up sources for each mode.

3.1 Configurations

3.1.1 System Control Register (SCR)

The SCR register controls features of entry to and exit from low power modes.

The selection of one of the four low power modes is controlled by the SLEEPDEEP bit in the Cortex-M3 System Control Register (SCR).

The bit assignments are shown in Fig 3.

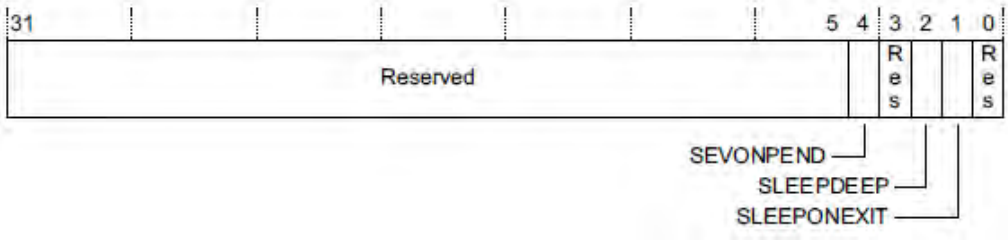

Fig 3. LPC1800 System Control Register (SCR, 0xE000ED10)

If the SLEEPONEXIT bit of the SCR is set to 1, when the processor completes the execution of an exception handler and immediately enters Sleep mode. This mechanism is useful for applications that only require the processor to run when an exception occurs.

If SLEEPON EXIT bit is set in the System Control Register, core will automatically enter the low power mode once the ISR has completed. This allows a low power application to be entirely interrupt-driven, so that the Cortex core will wake up, run the appropriate code and then re-enter the mode with minimal code being used for power management.

3.1.2 Power Management Controller (PMC)

The Power Management Controller (PMC) implements the control sequences to enable entering the different power modes and controls the power state of each peripheral. In addition, wake-up from any of the power-down modes based on hardware events is supported.

Power-down modes can be reached from active mode only and transitions between different Power-down modes are not allowed. Power-down modes are entered by a WFI or WFE instruction. Wake-up is caused by an interrupt or event. The interrupt cause is captured in the NVIC and events are captured in the Event router.

The PMC contains two registers, PD0_SLEEP0_HW_ENA and PD0_SLEEP0_MODE that select which of the low power modes to enter other than sleep mode.

3.1.3 Hardware sleep event enable register PD0_SLEEP0_HW_ENA

The PD0_SLEEP0_HW_ENA register must have the ENA_EVENT0 bit set in order to enter any of the low power modes other than sleep mode.

The bit assignments are shown in Fig 4.

Bit	Symbol	Description	Reset value	Access
0	ENA_EVENT0	Writing a 1 enables the Power-down modes for the Cortex-M3 (see the PD0_SLEEP0_MODE register for selecting the mode).	1	R/W
31:1	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	-	-

Fig 4. LPC1800 Hardware Sleep Event Register (PD0_SLEEP0_HW_ENA, 0x40042000)

3.1.4 Power-down modes register PD0_SLEEP0_MODE

The PD0_SLEEP0_MODE register controls which of the three Power-down modes, Deep-sleep, Power-down, or Deep power-down is entered when an ARM WFE/WFI instruction is issued and the SLEEPDEEP bit is set to 1.

The bit assignments are shown in Fig 5.

Bit	Symbol	Description	Reset value	Access
31:0	PWR_STATE	Selects between Deep-sleep, Power-down, and Deep power-down modes. Only one of the following three values can be programmed in this register: 0x0030 00AA = Deep-sleep mode 0x0030 FCBA = Power-down mode 0x0033 FF7F = Deep power-down mode	0x003F FF7F	R/W

Fig 5. LPC1800 Power Modes Register (PD0_SLEEP0_MODE, 0x4004201C)

3.1.5 Controlling power to the I/O pads

Bits 16:19 in the PD0_SLEEP0_MODE register control the power to the I/O block when entering a low power mode. If these bits are set, power to the I/O block is turned off upon entering the low power mode and the logic state of the pins is lost.

0x003F 00AA = Deep-sleep mode with I/O off

0x003F FCBA = Power-down mode with I/O off

0x003F FF7F = Deep power-down mode with I/O off

3.1.6 Programming the Clock Generation Unit (CGU)

Before entering Deep-sleep and Power-down modes, program the CGU as follows:

- Switch the clock source of all base clocks to the IRC.
- Put the PLLs in power-down mode.

Reprogramming the CGU avoids any undefined or unlocked PLL clocks at wake-up and minimizes power consumption during Deep-sleep mode.

3.1.7 Wait for Interrupt (WFI) instruction

Execution of the WFI instruction will cause immediate entry to any of the four low power modes based on the SLEEPDEEP bit, the ENA_EVENT0 bit, and the value of PWR_STATE bits.

The WFI instruction is a Cortex-M3 instruction which cannot be directly accessible by ANSI C. The CMSIS (Cortex Microcontroller Software Interface Standard) provides an intrinsic function to generate a WFI instruction and is supported by C compiler.

If a C compiler does not support the WFI intrinsic function, then the user will have to use inline assembler to access WFI instruction.

3.2 Programming steps to enter Sleep mode

The following steps must be performed to enter Sleep mode:

1. Set the SLEEPDEEP bit in the ARM Cortex-M3 SCR register to zero.
2. Use the ARM Cortex-M3 Wait-For-Interrupt (WFI) instruction.

Fig 6 below shows example code to enter sleep mode.

```
/* Release SLEEPDEEP bit in the System Control Register */
SCB->SCR &= (~(1 << 2));

/* Call ARM WFI function to enter low power state */
__WFI();
```

Fig 6. Code Example (Sleep mode)

3.3 Wake-up from Sleep mode

The MCU can wake up from Sleep mode through any enabled interrupt in the NVIC or any enabled event in the event router.

Sleep mode is exited automatically when an interrupt enabled by the NVIC arrives at the processor or a reset occurs. After wake-up due to an interrupt, the microcontroller returns to its original power configuration before entering Sleep mode. If a reset occurs, the microcontroller reboots and enters the default configuration in active mode.

3.4 Programming steps to enter Deep-sleep and Power-down modes

The following steps must be performed to enter Deep-sleep or Power-down mode:

1. Switch the clock source of all base clocks to the 12 MHz IRC.
2. Put the PLLs in power-down mode.
3. Set the SLEEPDEEP bit in the ARM Cortex-M3 SCR register to one.
4. Set the PD0_SLEEP0_MODE register to the Deep-sleep or Power-down value.
5. Use the ARM Cortex-M3 Wait-For-Interrupt (WFI) instruction.

Fig 7 below shows example code to enter Deep-sleep mode.

Fig 8 below shows example code to enter Power-down mode.

```
/* Set IRC as source clock for the core */
Chip_Clock_SetBaseClock(CLK_BASE_MX, CLKIN_IRC, true, false);

/* Power down the main PLL */
Chip_Clock_DisableMainPLL();
Chip_Clock_DisablePLL(CGU_USB_PLL);
Chip_Clock_DisableCrystal();

/* Set SLEEPDEEP bit in the System Control Register */
SCB->SCR |= (1 << 2);

/* Select Deep sleep as low power mode in PD0_SLEEP0_MODE register */
LPC_PMC->PD0_SLEEP0_MODE = (uinst32_t) 0x003000AA;

/* Call ARM WFI function to enter low power state */
__WFI();
```

Fig 7. Code Example (Deep-Sleep mode)

```
/* Set IRC as source clock for the core */
Chip_Clock_SetBaseClock(CLK_BASE_MX, CLKIN_IRC, true, false);

/* Power down the main PLL */
Chip_Clock_DisableMainPLL();
Chip_Clock_DisablePLL(CGU_USB_PLL);
Chip_Clock_DisableCrystal();

/* Set SLEEPDEEP bit in the System Control Register */
SCB->SCR |= (1 << 2);

/* Select Deep sleep as low power mode in PD0_SLEEP0_MODE register */
LPC_PMC->PD0_SLEEP0_MODE = (uinst32_t) 0x0030FCBA;

/* Call ARM WFI function to enter low power state */
__WFI();
```

Fig 8. Code Example (Power-down mode)

3.5 Wake-up from Deep-sleep and Power-down mode

The MCU can wake up from Deep-sleep mode and Power-down mode through any enabled interrupt in the NVIC or any enabled event in the event router.

Wake up from Power-down mode needs extra care if the internal SRAM is being used for code or data storage as only the top 8 kB of the SRAM starting at memory region 0x1008 8000 is retained.

3.6 Programming steps to enter Deep Power-down mode

The following steps must be performed to enter Deep Power-down mode:

1. Set the SLEEPDEEP bit in the ARM Cortex-M3 SCR register to one.
2. Set the PD0_SLEEP0_MODE register to the Deep Power-down value.
3. Use the ARM Cortex-M3 Wait-For-Interrupt (WFI) instruction.

Fig 9 below shows code examples to enter Deep Power-down mode.

```
/* Set SLEEPDEEP bit in the System Control Register */
SCB->SCR |= (1 << 2);

/* Select Deep Power-down as low power mode in PD0_SLEEP0_MODE register */
LPC_PMC->PD0_SLEEP0_MODE = (uinst32_t) 0x0033FF7F;

/* Call ARM WFI function to enter low power state */
__WFI();
```

Fig 9. Code Example (Deep Power-down mode)

3.7 Wake-up from Deep Power-Down mode

When the LPC18x0 wakes up from Deep power-down mode, the boot loader configures the PLL1 as the clock source running at 96 MHz and attempts to boot similar to a reset or power-up.

4. Additional tips to reduce power consumption

On the LPC18x0 series, the current consumption can be further reduced by considering the following:

4.1 Software

Most embedded applications terminate with a while(1) loop, and they service interrupts whenever needed. In this case, code is still constantly fetched from the on-chip flash and executed which adds to the power consumption. A better solution would be to switch to a power saving mode like sleep mode and then wait for interrupts. An interrupt from a peripheral would then wake the device from sleep mode. When the MCU is running at 180MHz with all peripherals on, as much as a 20% current reduction can be achieved while busy waiting in the while(1) loop in Sleep mode.

4.2 CPU clock rate

On the LPC18x0, the CPU clock rate can be controlled as needed by changing clock sources, reconfiguring PLL values, and/or altering the CPU clock divider value. This allows a trade-off between power and processor speed based on application requirements. Lowering the PLL0's output frequency can also save power.

4.3 Port pins

The general purpose port pins on the LPC18x0 have programmable internal pull-ups. Additional steps should be considered to reduce current consumption in the low power modes by configuring the port pins appropriately to prevent contention. It is recommended to disable all internal pull-ups and pull-downs.

Please note that in deep power-down mode, the state of the port pins does not affect the current consumption and the steps mentioned above do not need to be considered in this mode.

4.4 Unused peripherals

When not using certain peripherals, the clock generation unit (CGU) can be programmed to shut down the clocks to unused peripherals in order to save power.

5. Low power mode demos

5.1 Objective

This application note provides four low power mode examples and is based on NXP's LPCOpen software platform. The examples support three IDE tools (LPCXpresso, Keil, and IAR).

To reprogram the device, the user must put the device in ISP mode by power cycling the board while pushing down the ISP button or pulling P2_7 low. Thereafter, the device can be erased and programmed again.

The executable codes are required to be programmed on the external quad SPI flash connected to SPIFI interface since there is no on-chip flash on LPC18x0 (flashless part).

5.2 Requirements

1. Keil MDK, IAR EWARM or LPCXpresso.
2. Keil MCB1800 Evaluation Board with a socket for alternative MCU chip (see Fig 10).

Remark: The part number “LPC1850FET256” is measured on the boot ROM version of V11.2 in this application note. And two parts with the same boot ROM version are used for testing.

5.3 Keil MCB1800 Evaluation Board

5.3.1 Hardware setup

The software example provided with this application note is based on LPCOpen software framework and is intended to run on the Keil MCB1800 evaluation board. To boot from the external quad SPI flash, the boot jumper for P1_1 pin must be short to “H” and other boot pins P1_2, P2_8 and P2_9 pins must be short to “L”. To communicate using a PC Terminal, UART3 TX and RX must be enabled through the J16 and J13 jumpers respectively. To wake up from a low power mode, the WAKEUP0 button must be pushed (see Fig 10).

5.3.2 Measuring current consumption

The Keil MCB1800 board gives the customer access to the chip’s core voltage, labeled as 3V3R on the J1 jumper (see Fig 10). By connecting an ammeter between the 3V3R pins, the current consumed by the chip can be monitored at all times.

5.3.3 Measuring the Wake-Up time

In order to measure the wake-up time taken by MCU, the easiest way is to use the WAKEUP0 pin to wake up the MCU from power down modes. When the WAKEUP0 signal is asserted, the MCU will wake up and enter the event router interrupt handler and toggle a GPIO pin. This GPIO pin is set to a logic `'0'` in `main()` before going into sleep and set to a logic `'1'` after the MCU wakes up. In this way, the time between the assertion of the WAKEUP0 signal and rising edge of the GPIO pin will be the approximate wake-up time taken by MCU. The WAKEUP0 pin and GPIO that is toggled are shown in Fig 10 below. Please note that this method of measuring the wake-up time includes the Cortex-M3 interrupt latency of 12 clock cycles in addition to the time it takes to toggle the GPIO.

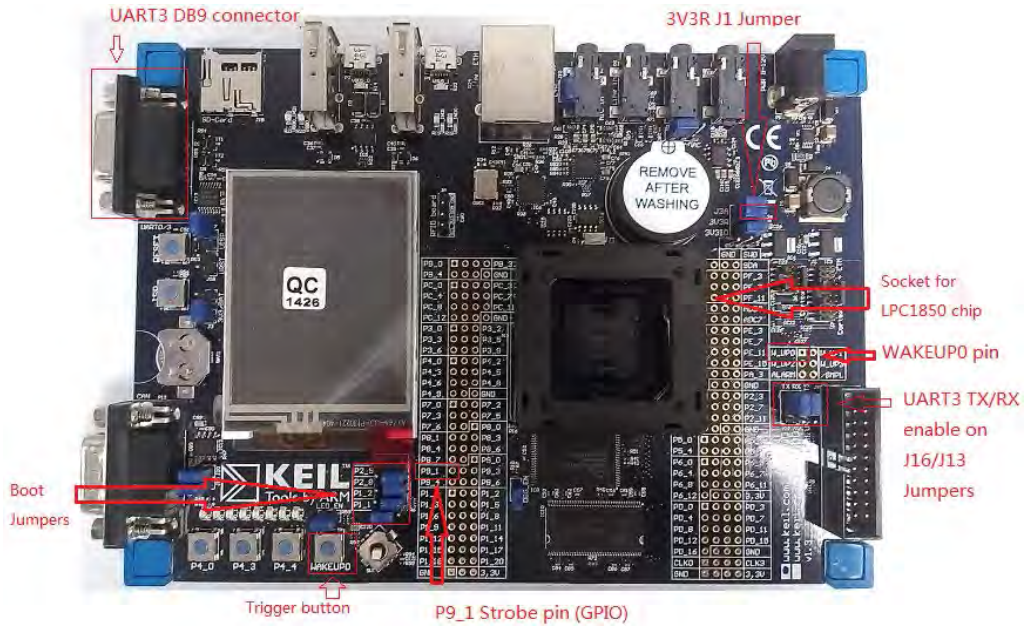

Fig 10. Keil MCB1800 Evaluation Board with a socket for LPC1850

5.3.4 Software Example

In active mode, the LPC18x0 is running at 12MHz (Internal RC Oscillator).

1. Connect the USB cable to power up the Keil MCB1800 evaluation board via micro USB0/1 port.
2. Connect the DB9 cable and open a terminal emulator on the corresponding COM port with a baud rate of 115200.
3. Open the project file included in any of the three supported IDEs. Compile and download the code to external Quad SPI flash.
4. Power cycle the MCB1800 to execute the code. LED6 (on port pin P9_1) should be off.
5. On the terminal emulator, enter a number between one through four to enter one of the four low power states (see Fig 11).

```
================================================================================
NXP Semiconductors LPC18xx Low Power Application Note Software Example
================================================================================
Select the power state test to be executed
 - Sleep state & Wake up test           : Press '1' to select
 - Deep Sleep state & Wake up test      : Press '2' to select
 - Power down state & Wake up test      : Press '3' to select
 - Deep power down state & Wake up test : Press '4' to select
================================================================================
```

Fig 11. Terminal display

6. Upon choosing a low power state, you will be able to choose one of two wake-up mechanisms: WAKEUP0 pin or RTC alarm. Choose WAKEUP0 to measure the wake-up time.
7. The MCU is now in the low power mode chosen. The current reading on the ammeter is the current consumed in the current low power state. The wake-up time can be measured by using the following pins:
   a. Trigger pin (WAKEUP0) – Used to get the device out of the low power modes (sleep, deep-sleep, power down). The pin is triggered externally low (falling edge) to wake the device up. Use the push button shown in Fig 10.
   b. Strobe pin (P9_1) – After wake-up the device returns to active mode and will set this pin high from within the EVRT_IRQHandler() subroutine.
8. The wake-up time is the difference between the falling edge of the trigger pin and the rising edge of the strobe pin. See Fig 12 and Fig 13 . This applies when measuring sleep mode, deep-sleep mode, and power down mode. For deep-power down mode, the MCU goes through a reset process and the strobe pin will be pulled down once the MCU completes the reset process and starts executing user code (see Fig 14).

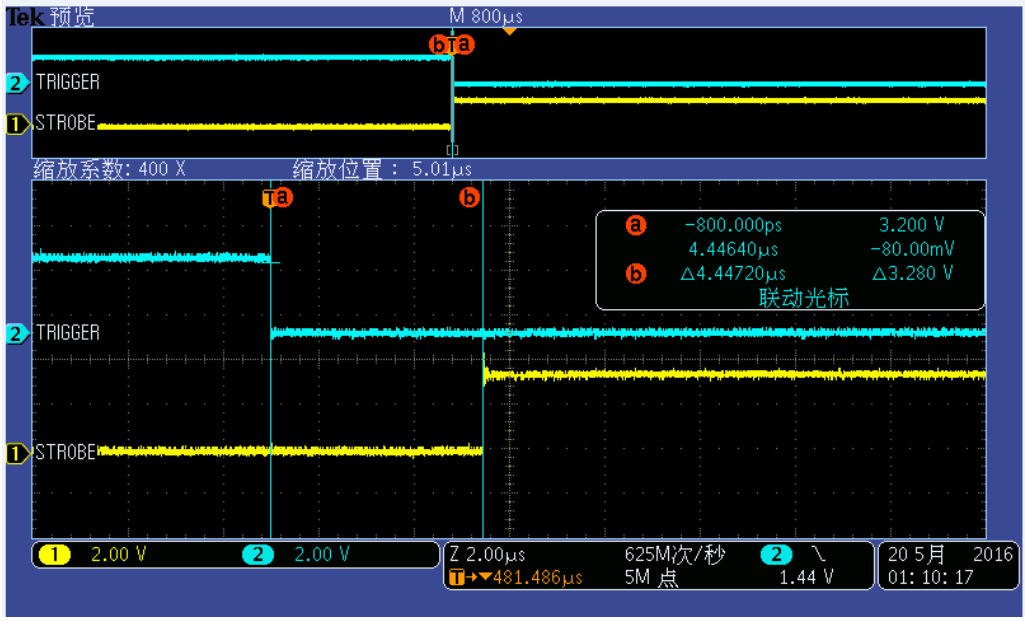

Fig 12. Wake-up time measurements (sleep)

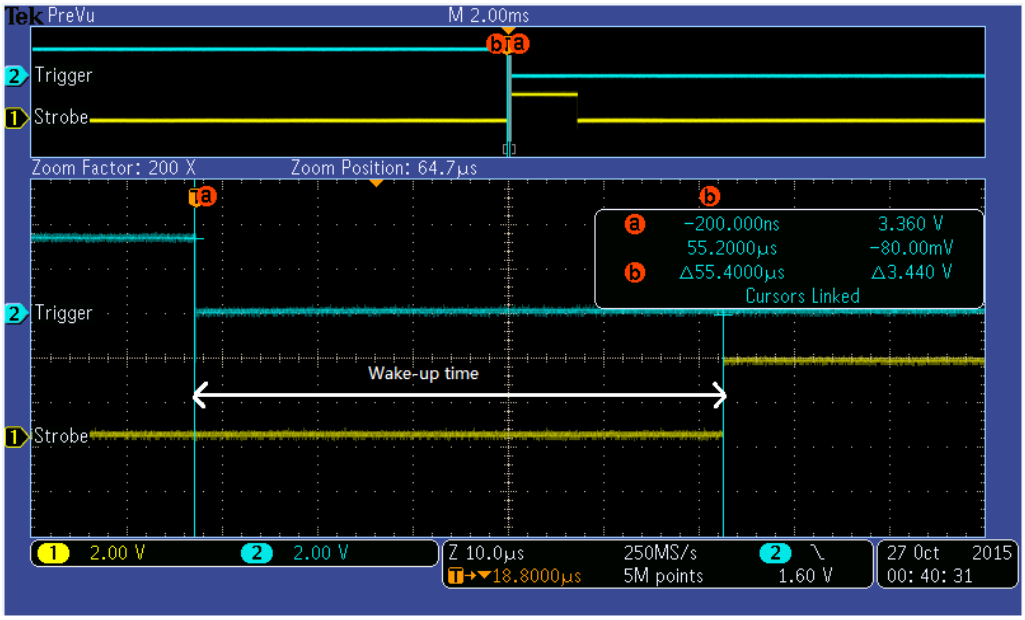

Fig 13. Wake-up time measurements (deep-sleep, power-down)

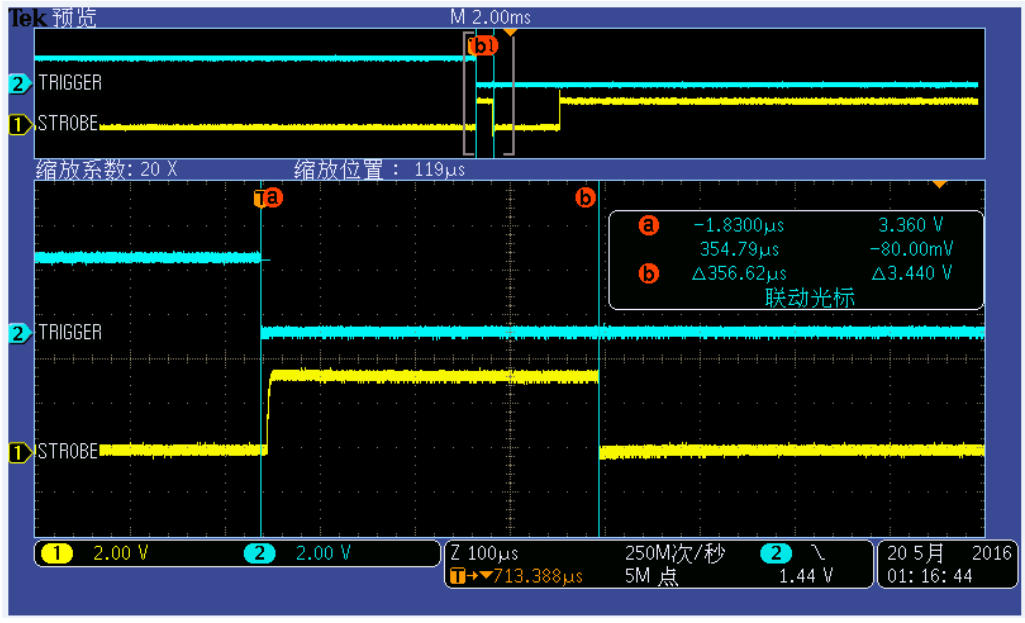

Fig 14. Wake-up time measurements (deep power-down)

5.3.5 Power measurements and Wake-up measurements

Table 1 shows the typical power consumption and wake-up measurements for the particular Keil MCB1800 board used for testing:

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

Low power modes	Ivdd current	Wake-up time (QSPI flash)	Wake-up time (SRAM)
Sleep mode (IRC (12 MHz), all peripherals off except SPIFI, UART3 and GPIO)	7.5 mA	4.4 µS	2.2 µS
Deep-sleep Mode	60 µA	55.4 µS	52.4 µS
Power-down Mode	18.3 µA	55.4 µS	52.4 µS
Deep Power-down Mode	[1] 2.2 µA[2] 0.05 µA	356.6 µS	–

[1] VBAT floating
[2] VBAT = 3.6V

5.3.6 Debug Notes

The user should be aware of certain limitations during debugging. It is recommended not to use Deep-sleep, Power-down or Deep power-down during a debug session. Once an application is downloaded via JTAG/SWD interface, the USB to SWD/JTAG debug adapter (Keil ULINK2 for example) should be removed from the target board, and then the LPC18x0 power cycled to allow wake-up from Deep-sleep, Power-down and Deep power-down modes. Another issue is that debug mode changes the way in which reduced power modes are handled by the Cortex CPU. This causes power modes at the device level to be different from normal modes operation. These differences mean that power measurements should not be made while debugging as the results will be higher than during normal operation in the application.

5.3.7 Waking up to SRAM from Power-down mode

For some applications, it may be necessary to save as much power as possible by implementing a duty-cycle approach of operation. This means the MCU will be in a low power state whenever it is idle, and will periodically turn itself on to perform some work. In order to maintain flexibility in the frequency of this duty-cycle approach of operation, it is important to wake up as fast as possible.

As shown in Table 1, Power-down mode consumes the least amount of power without resetting the MCU, and can be programmed to wake up to SRAM in order to decrease the wake up time. In `pmc_states.c` there is a macro called `PowerDownToSRAM` that can be turned on in order to wake-up to SRAM from Power-down mode.

6. Conclusion

The LPC1800 series provides great flexibility and various options for users to achieve low power consumption at various wake-up times. As shown in Table 1, the user has four low power mode options to choose from in order to achieve the desired power consumption. This flexibility allows a trade-off between power consumption and wake-up time based on the user's application requirements. In addition, the user has the option of using the self-wake-up timer (for example, RTC) feature and would not require an external component to wake up the device.

7. Legal information

7.1 Definitions

Draft — The document is a draft version only. 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 herein and shall have no liability for the consequences of use of such information.

7.2 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.

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.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

7.3 Trademarks

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

8. Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 12 August 2016
Document identifier: AN11853