AN11332

LPC800 I2C wake-up from power-down mode

Rev. 1.1 — 13 March 2013

Application note

Document information

Info	**Content**
Keywords	LPC800, wake-up, power-down, I2C slave
Abstract	The LPC800 family of microcontrollers allows wake-up from the power-down state using an I2C interrupt. This application note details the wake-up procedure and the behavior of the LPC800 to these wake-up events.

Revision history

Rev	Date	Description
1.1	20130313	Corrected cross-reference links.
1	20130227	Initial version.

Contact information

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

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

1. Introduction

The LPC800 family of microcontrollers incorporates four low-power modes:

1. Sleep mode: affects the ARM Cortex-M0+ core only. The clock to the core is disabled, but the peripherals and memories remain active.
2. Deep-sleep: the peripherals receive no internal clocks. Memories are in stand-by mode. All registers maintain their internal states. The Windowed Watchdog Timer (WWDT), Wake-up Timer (WKT), and Brown-out Detection (BOD) can remain active to wake up the system using an interrupt.
3. Power-down: the peripherals receive no internal clocks. All registers maintain their internal states. The flash memory is powered down. The WWDT, WKT, and BOD can remain active to wake up the system on an interrupt.
4. Deep power-down: For maximal power savings, the entire system is shut down except for the general-purpose registers in the PMU and the self wake-up timer. All registers maintain their internal states. The part can wake up on a pulse on the WAKEUP pin or when the self wake-up timer times out. On wake-up, the part reboots.

The ability to wake-up from power-down from an I2C, SPI, or USART wake-up event is a new and really useful feature in the LPC800 family.

Table 1 shows that in power-down mode, almost everything is turned off. The only peripherals that could be active in this low power mode are the brown-out detection circuit, the Windowed Watchdog Timer, and the Wake-up Timer. The default state for these peripherals is off. This allows the LPC800 to achieve a current consumption of only 1.8 µA in power-down mode.

Table 1. Peripheral configuration in reduced power modes

Peripheral	Sleep mode	Deep-sleep mode	Power-down mode	Deep power-down mode
IRC	software configurable	on	off	off
IRC output	software configurable	off	off	off
Flash	software configurable	on	off	off
BOD	software configurable	software configurable	software configurable	off
PLL	software configurable	off	off	off
SysOsc	software configurable	off	off	off
WDosc/WWDT	software configurable	software configurable	software configurable	off
Digital peripherals	software configurable	off	off	off
WKT/low-power oscillator	software configurable	software configurable	software configurable	software configurable

2. Procedure for waking using an I2C interrupt

Waking up the LPC800 from power-down mode requires three steps:

1. Enable the I2C interrupt. This can be accomplished using the NVIC_EnableIRQ(I2C_IRQn) function in the I2C_SlvInit function.

2. Set bit 8 in the Start logic 1 interrupt wake-up enable register (STARTERP1), as shown in Table 2.
3. Configure the I2C peripheral in slave mode

Table 2. Start logic 1 interrupt wake-up enable register(STARTERP1, address 0x4004 8214) bit description

Bit	Symbol	Value	Description	Reset Value
0	SPI0		SPI0 interrupt wake-up	0
		0	Disabled	
		1	Enabled	
1	SPI1		SPI1 interrupt wake-up	0
		0	Disabled	
		1	Enabled	
2	-		Reserved	-
3	USART0		USART0 interrupt wake-up. Configure USART in synchronous mode.	0
		0	Disabled	
		1	Enabled	
4	USART1		USART1 interrupt wake-up. Configure USART in synchronous mode.	0
		0	Disabled	
		1	Enabled	
5	USART2		USART2 interrupt wake-up. Configure USART in synchronous mode.	0
		0	Disabled	
		1	Enabled	
7:6	-		Reserved	-
8	I2C		I2C interrupt wake-up.	0
		0	Disabled	
		1	Enabled	
11:9	-		Reserved	-
12	WWDT		WWDT interrupt wake-up.	0
		0	Disabled	
		1	Enabled	
13	BOD		BOD interrupt wake-up.	0
		0	Disabled	
		1	Enabled	
14	-		Reserved	-
15	WKT		WWDT interrupt wake-up.	0
		0	Disabled	
		1	Enabled	
31:16	-		Reserved	-

All information provided in this document is subject to legal disclaimers.
© NXP B.V. 2013. All rights reserved.

3. Wake from power-down mode example code

The LPC812 LPCXpresso board is used to demonstrate wake-up from power-down mode. The evaluation board details can be found at: http://www.nxp.com/demoboard/OM13053.html. The application code example provided with the AN operates as described below.

While the device is in active mode, the on-board LED blinks at a 1 Hz rate. The LED is connected to port PIO0_7.

Configuring the LED output is simply a matter of setting PIO0_7 to output mode:

```
LPC_GPIO_PORT->DIR0 |= (1<<7);
```

The Systick timer is programmed with a 0.5 Hz rate:

```
SysTick_Config( SystemCoreClock / 2 );
```

The Systick timer interrupt uses one line of code to toggle the PIO0_7 port pin:

```
void SysTick_Handler(void) {
  LPC_GPIO_PORT->NOT0 = (1<<7);
}
```

The Systick timer cannot wake the LPC812 from power-down mode, since there are no clocks active to drive the timer.

When the LPC812 is in power-down mode, the LED stops blinking, but should start again when it receives its slave address. The slave address can be changed by the define statement in the lpc8xx.h file:

```
/* When the I2C is a slave, SLAVE_ADDR is the slave address. */
#define SLAVE_ADDR    0x54
```

In the example code, the command to place the LPC812 into power-down is the I2C slave address 0x54 followed by data byte 0xAA, however, the user can use other methods to place the device into power-down mode. Wake-up occurs when the LPC812 receives its slave address on the bus.

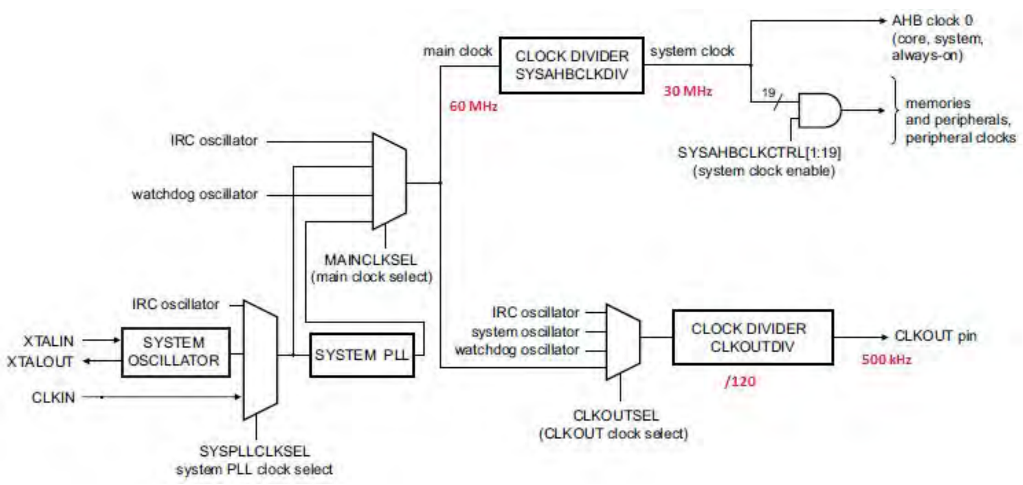

Fig 1. System clock and CLKOUT generation

The CLKOUT feature is used to show when the device has been shutdown and woken up. The system clock is running at 30 MHz. The CLKOUT is driven by the output of the PLL which is running at 60 MHz, so we set the clock output divider (CLKOUTDIV) to 120 to achieve a 500 kHz CLKOUT signals, as shown in Fig 1.

4. Wake from I2C Interrupt

Demonstrating a wake from an I2C interrupt is simply a matter of enabling the wake signal in the STARTERP1 register:

```
/* Enable wake from power down using I2C */
LPC_SYSCON->STARTERP1 |= (1<<8);
```

The I2C interrupt must be enabled, and this is accomplished in the I2C_SlvInit function in the lpc8xx_i2c.c file.

4.1 Normal I^{2}C-bus behavior

When an I2C master transmits data to an LPC800 which is operating in active mode, you will see its behavior in the scope trace, shown in Fig 2.

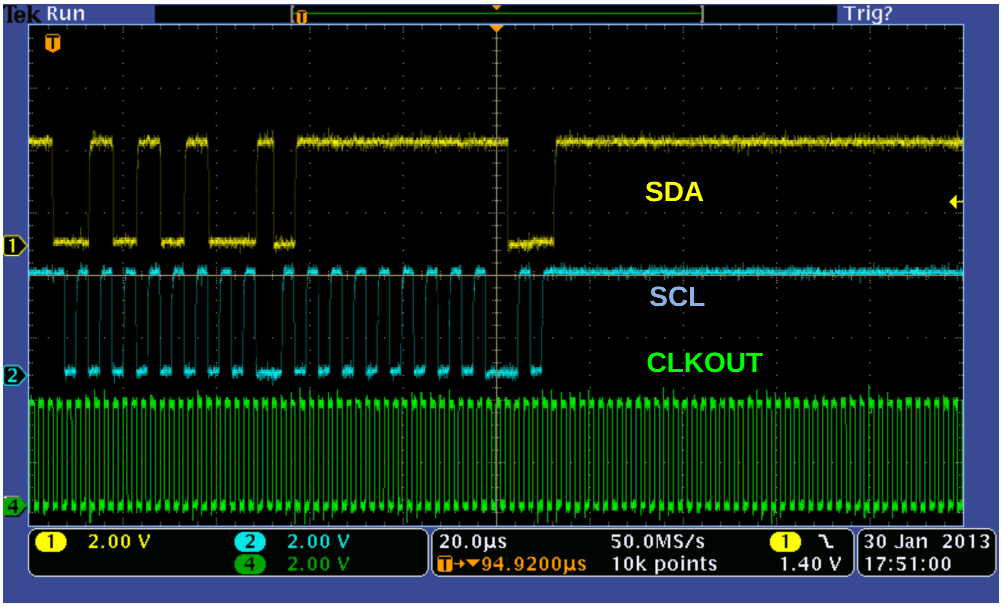

Fig 2. Normal I2C operation

The top trace is SDA (I2C data); the middle trace is SCL (I2C clock). These signals are generated by the I^{2}C-bus master, with the exception of the acknowledge phase where the LPC800 pulls the SDA line low. The bottom trace is the 500 kHz CLKOUT signal from the LPC800.

Note that the LPC800 holds the clock line low for a short period of time before it pulls the SDA line low to acknowledge its slave address. This small amount of clock stretching can be seen between the 8th and 9th clock pulses, as well as between the 17th and 18th clock pulses. Clock stretching occurs whenever there is an acknowledge phase on the bus.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2013. All rights reserved.

The I^{2}C-bus specification allows a slave device to stretch the clock, which is similar to inserting a wait state on the bus. The I^{2}C-bus specification does not specify a maximum length of the clock stretching event.

4.2 Power-down behavior

In order to demonstrate the power-down feature, the I2C slave device is configured to respond to address 0x54, and when it receives a data byte 0xAA, it will enter the power-down mode.

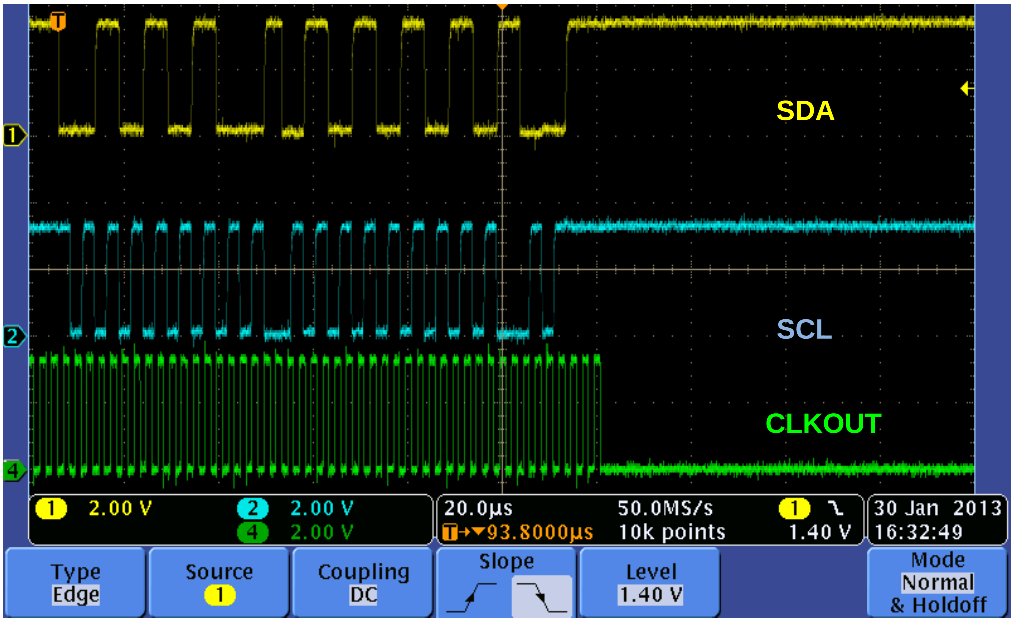

Fig 3. Power-down command

In Fig 3, you can see that the LPC800 responds to the slave address 0x54 by acknowledging the address. The I2C master then a data byte with a value of 0xAA. After the Stop condition has been sent, the LPC800 checks the data byte, and if it detects 0xAA, it will go into power-down mode.

In power-down mode, the internal RC (IRC) oscillator, flash, PLL, and the system oscillator are turned off. We are using the IRC oscillator for this example, and you can see that after the I2C Stop condition has been sent, the CLKOUT signal no longer toggles, and goes low, indicating that it has entered the power-down mode.

4.3 Wake from power-down

The LPC800 has been configured to wake when an I2C interrupt occurs. An interrupt occurs when the LPC800 detects its own address on the bus. If an address is sent that does not match its I2C slave address, no interrupt is generated, so it will remain in power-down mode.

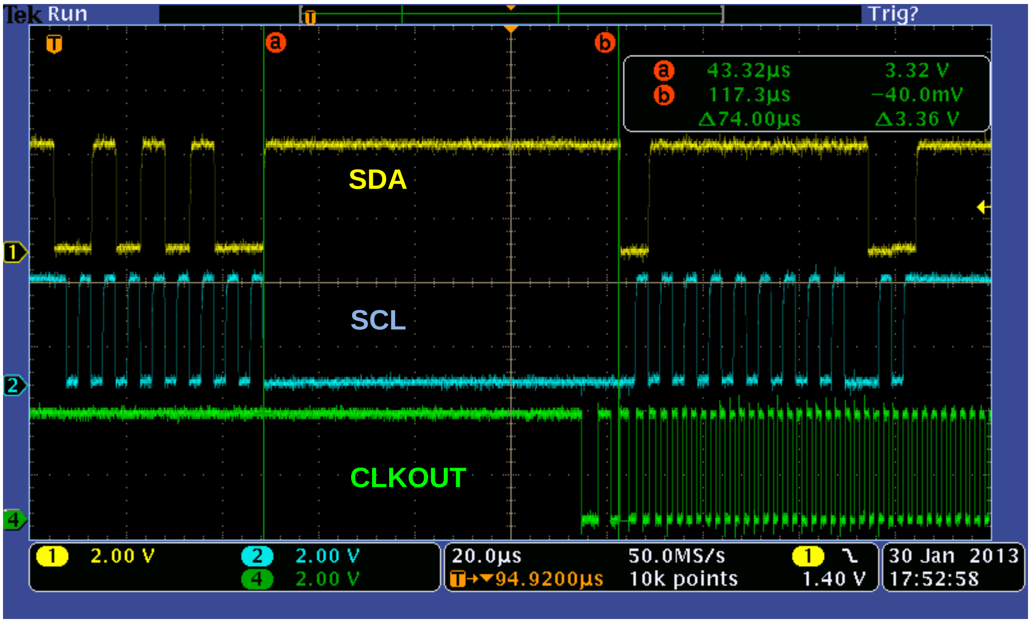

Fig 4. Wake from power-down with IRC oscillator

In Fig 4, you can see that after the LPC800 receives its slave address, it will hold the SCL line low until the device has powered back up, and is receiving clock signals from the internal RC oscillator. As the system powers up, and the oscillator starts, you can see that it takes a while for the oscillator to settle to its operating frequency and the PLL to lock.

As soon as the LPC800 is operational, it will enter the interrupt service routine and pull the SDA line low to acknowledge its address. It will then release the clock line, so the I2C master can once again start sending clock pulses.

In this example, it takes approximately 74 microseconds from the rising edge of the SDA signal (the I2C read/write bit), which is where the I2C master has released the SDA line, to the falling edge of the acknowledge signal (where the slave has started to control the data line).

Also note that the behavior of the I^2C-bus is back to normal after the clock stretching event, and further clock stretching is not required, except for the normal clock stretching observed when the acknowledge is transmitted.

4.4 Wake from power-down using crystal oscillator

The previous example code showed the wake-up triggered by an I2C interrupt, while using the internal RC oscillator as the clock source to the PLL.

All information provided in this document is subject to legal disclaimers.
© NXP B.V. 2013. All rights reserved.

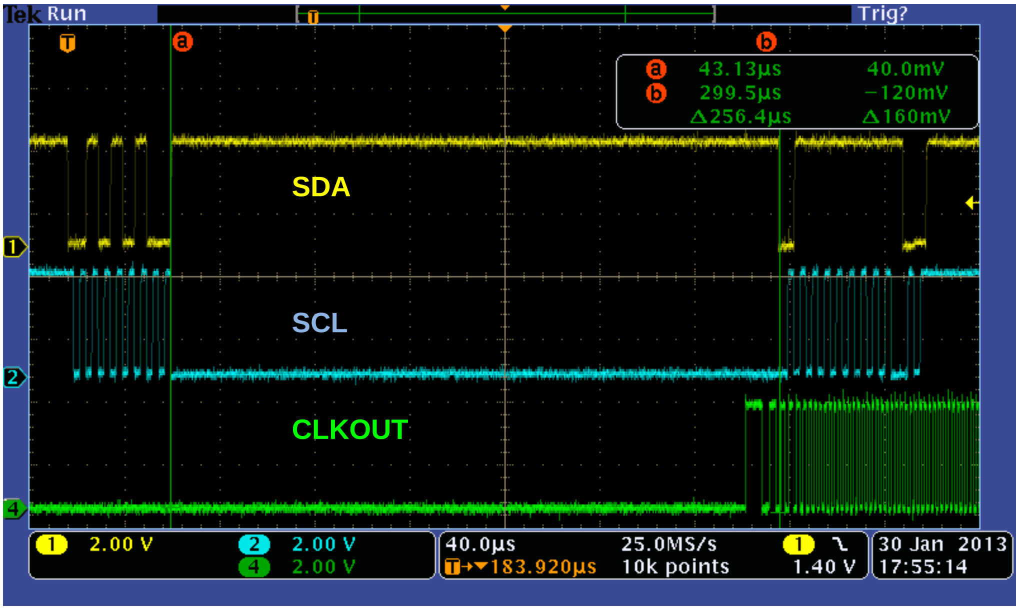

Fig 5. Wake-up from power-down using with a crystal oscillator

If we reconfigure the system to use the crystal oscillator to feed the PLL, we would expect that the start-up time would be longer than the internal oscillator.

Fig 5 shows the time required for the system to respond to an I2C interrupt. It can be seen that the wake-up time has increased significantly. The time between the falling edge of the eighth slave address bit and the falling edge of the acknowledge signal was measured at over 250 microseconds, compared to 74 microseconds for the internal oscillator.

5. Conclusion

The LPC800 has the ability to wake-up from a power-down state where no clock to the core or peripheral is required. This allows the microcontroller to consume very little power, while still remaining responsive to external signals, such as I2C.

It is important to understand the behavior of the LPC800 when operated as an **I2C slave** in power-down mode, especially the clock stretching that occurs as the device moves from the power-down state to an active state. If these are taken into consideration, a user should be able to place the device in a very low power state, which would require a higher power mode in other microcontroller families.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2013. All rights reserved.

6. Legal information

6.1 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.

6.2 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.

6.3 Trademarks

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

7. List of figures

8. List of tables

9. Contents

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

© NXP B.V. 2013. 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: 13 March 2013
Document identifier: AN11332