# AN10493

## LPC214x power-down mode and USB wake-up

Rev. 01 — 27 June 2006 Application note

### Document information

| Info | Content |
|---|---|
| **Keywords** | Microcontroller, MCU, Power-down mode, Wakeup, BOD. |
| **Abstract** | This document describes how to put the MCU into Power-down mode to save power and provides one example to wake up the MCU via USB activity. |

## Revision history

| Rev | Date | Description |
|---|---|---|
| 01 | 20060627 | Initial revision |

# Contact information

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

For sales office addresses, please send an email to: sales.addresses@www.semiconductors.philips.com

# 1. Introduction

The family of LPC2141/2/4/6/8 microcontrollers supports two reduced power modes: Idle mode and Power-down mode. In Power-down mode, the oscillator is shut down and the chip receives no internal clocks, but the MCU state, peripheral register values, and the value of the internal SRAM are preserved. The power consumption in the Power-down mode is minimum, and the supply current can drop up as low as 30 µA when BOD is disabled. The system can wake up from Power-down mode using external interrupts, the RTC clock, the BOD interrupt, and/or activities on the USB bus.

This document describes how to make hardware modifications to minimize the power consumption on the Keil's MCB2140 board. It also provides the necessary software configuration to put the MCU into Power-down mode and then use USB bus activity to wake up the MCU.

The hardware platform is an MCB2140 board from Keil, now an ARM company. The software is primarily based on Keil's LPC2148 USB Mass Storage Class Example. Some software logic is added in the interrupt handler to handle the USB suspend and resume process, the initialization routine, and finally, the endless main loop, which checks the USB status, handles entry into power down, and exits from USB wake-up.

# 2. Power-down mode and USB wake-up

This section describes how to make the hardware and software changes to put LPC214x into Power-down mode and then configure the interrupt wake-up register such that any activity on the USB bus can wake up the MCU.

## 2.1 Hardware configuration

The MCB2140 Evaluation Board was developed by Keil. It ships with an LPC214x device that is a superset of several other device variants of the Philips LPC2100 microcontroller series. The MCB2140 board contains all the hardware components that are required in a single-chip LPC2100 system and have a USB interface for both power and serial communication.

The Keil MCB2140 Evaluation Board connects to your PC using the serial port (for Flash download using the Philips LPC210x FLASH ISP Utility) or the JTAG interface (for program debugging using the Keil ULINK USB-JTAG Adapter and the µVision IDE and Debugger).

To minimize the power consumption, the hardware modifications that need to be done on the board are as follows:

- Port 0:
  - All port 0 must be terminated. If left as inputs, they will float and draw current from the pad ring. In the software, we simply set them to outputs and drive them high. Make sure that there will be no load on these pins, or else current will flow from 3.3V $V_{CC}$ supply through the load.

- Port 1:
  - Remove all pull-down resistors on port 1. All pins on port 1 have internal pull-ups. If grounded, each pin draws about 30µA from the 3.3 V $V_{CC}$ supply. In software, all the port 1 pins are set to inputs.
- Debug pins:
  - R28 on JTAG TCK pin is removed.
  - ETM, JTAG jumpers are removed. JTAG jumper is installed during flash programming and removed after finishing.
- USB pads:
  - D+ signal is pulled high on the board as the full-speed indicator.
  - D- signal needs to be terminated if the USB cable is not connected to the host. A 100k pull-down resistor is added on D- signal.
- Unused jumpers are removed, including VR, V3A, AOUT, RST, VREF, and RST.

## 2.2 Software for Power-down and USB wake-up

### 2.2.1 Overview

Based on USB Specification 2.0, Chapter 7.2 “Power Distribution”, and Table 7.7 “DC Electrical Characteristics”,

- in any powered state, the device power allocation should not exceed 100mA.
- In suspend state, the device power allocation should not exceed 500 µA.

Excluding many on-chip peripherals on LPC2148, the MCU, the USB pull-up, discrete components such as the UART transceiver, the audio amplifier, and the use of PLL0 for the system clock (CCLK) and PLL1 for the USB device contribute a significant portion of the power consumption of the MCB2140 board.

In order to meet the power consumption requirement:

- The system clock has been reduced to 24MHz via PLLCFG register.
- The peripheral clock (PCLK) has been reduced to 24MHz, same as the CCLK, via VPBDIV register, since the USB block in LPC214x requires that the PCLK should be greater than 18MHz.

The overall software initialization sequence and program flow should include following steps:

- Setup PLL clock configuration and VPBDIV register.
- Initialize VIC, all the ports, and USB block.
- Install USB interrupt handler and enable USB interrupt.
- Setup interrupt wake up source in INTWAKE register.
- Enter endless loop wait for USB suspend and Resume interrupts. If the USB suspend interrupt occurs and USB_needclk is low, the board will go to power down mode by setting PCON register. If the USB resume interrupt occurs, the board will wake up from power down mode, the software will reconfigure the PLL, reinitialize the USB block, and finally resume normal operation.

### 2.2.2 Clock configuration

The CCLK of Keil MCB2140 board is set to 24MHz when PLL0CFG = 0x21. The VPB clock, PCLK, is set to 24MHz when VPBDIV = 1. The USB clock, PLL1, is set to 48Mhz, where FOSC = 12MHz, M = 4, P = 2, PLL1CFG = 0x23.

Set the INTWAKE register (INTWAKE = 0x20) so that any activity on the USB bus will wake up the MCU from Power-down mode.

### 2.2.3 Initialization

The pseudo code to handle MCU from normal operation to power-down, then from power-down to normal operation is shown below:

1. Set all I/O pins to GPIO, port 0 to output high, port 1 to input.

   **Note:** since port 2 is set to GPIO, the DEBUG feature will be disabled. To program the flash using the Philips ISP flash utility, press INT1 button and hold to drive P0.14 low during the process. The alternative is to enable DEBUG bit on port 2 during debugging and disable debugging for final release and measurement.

2. Set up a timer to add some delay before going to Power-down mode.
3. USB reset and initialization: enable USB interrupts, most importantly the DEV_STAT bit in the USB device interrupt enable register.
4. Send SET_MODE command to USB protocol engine to clear the AP_CLK bit.

   **Note:** By default, the AP_CLK bit is cleared already. If this bit is set, the USB_needclk will always be 1, and the MCU can never be set to Power-down mode. The USB_needclk status bit can be found in USB Interrupt Status register.

5. Set USB wake bit in interrupt wake-up register, INTWAKE = 0x20.

### 2.2.4 Interrupt handler

```
Int USB_Interrupt_Handler ()           __irq
{
if ( USBDevIntSt & DEV_STAT )
           {
                 /*Set DEV_STAT bit in USBDevIntClr to clear the interrupt
                 Send GET_DEV_STATUS command to USB protocol engine*/
                 if ( DEV_SUS_CH is true )        /* Suspend or Resume */
{
if ( DEV_SUS is true )                 /* Suspend */
{
set USB Suspend flag
start the timer
}
else                                              /* Resume */
{
clear USB Suspend flag
stop the timer
}
                 }
}
}
```

### 2.2.5 Main routine

/* Endless loop from normal operation to Power-down mode, and then wake up when activities on USB bus is seen. */

```
While ( 1 )
{
/* The following if statement is always false unless the MCU has been in the
Power-down mode already and waken by an external interrupt. */
     if ( power_down flag is set )    /* wake up from power down mode */
{
           clear power_down flag
/* Power-down mode automatically turns off and disconnects activated PLL(s).
Wakeup from Power-down mode does not automatically restore the PLL settings. This
must be done in software. */
           reset main PLL0 for system clock
           reset PLL1 for USB clock
           re-initialize and reset USB block
}

/* if device has been suspended for a short period of time and USB_needclk is
zero, we can set the MCU to Power-down mode to preserve the power consumption.
The status of the USB_needclk is in the USB Interrupt status register, USBIntSt.
*/

if ( USB suspend flag is true && USB_needclk bit in USB Interrupt Status register
is low )
{
        If ( a fixed grace period is met )
        {
            USB suspend flag is false.  /* Reset its flag before waken up. */
            power_down flag is set          /* used when wakeup occurs */
PCON = 0x06;        /* The BOD circuit is disabled during Power-down mode. */
        }
    }
}
```

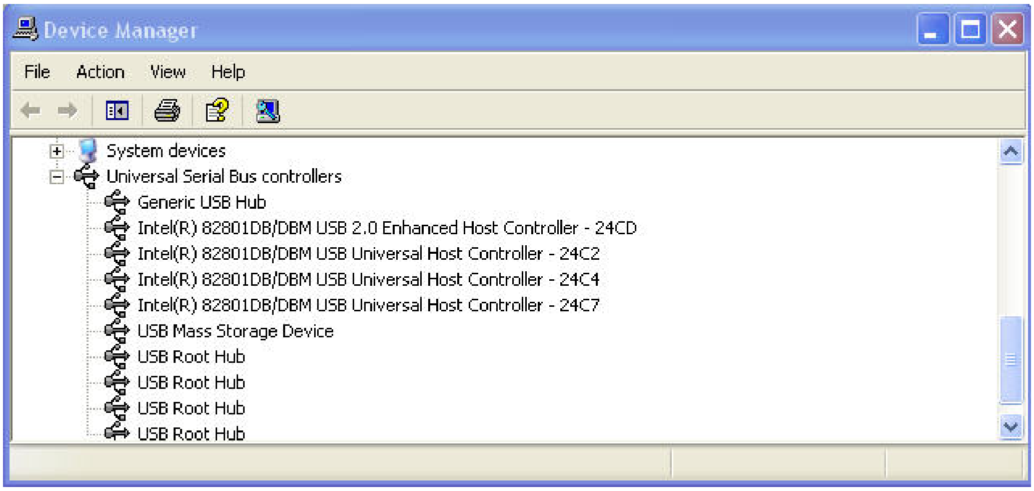
## 3. Host operation and measurement

To monitor the current drop, take the jumper J3 (3.3V) out, use a multi-meter to connect on each end, and measure the current.

Use a USB cable to connect between a PC host and MCB2140 board. When the USB power is applied on the board, the current measurement will be in the 30mA range.

The Windows operating system will show a **"Found New Hardware"** message. You won't have to load any special drivers. Because support is built into Windows 2000 and Windows XP, a standard Mass Storage Device does not require a special USB driver. A few seconds later, a new disk drive should appear on the Windows screen.

Open the Windows Hardware Device Manager, and a “USB Mass Storage Device” will appear under Universal Serial Bus Controller. Then right click “USB Mass Storage Device”. You can click “Enable” to resume or “Disable” to suspend USB operation. When, the USB device is suspended, the current measurement will drop significantly from 30mA down to 50µA or even lower after a short delay. The MCU now is in the power down mode. When the USB device is resumed, the current measurement will rise back to 30mA, and the new disk drive will reappear on the Windows screen.



- System devices
- Universal Serial Bus controllers
  - Generic USB Hub
  - Intel(R) 82801DB/DBM USB 2.0 Enhanced Host Controller - 24CD
  - Intel(R) 82801DB/DBM USB Universal Host Controller - 24C2
  - Intel(R) 82801DB/DBM USB Universal Host Controller - 24C4
  - Intel(R) 82801DB/DBM USB Universal Host Controller - 24C7
  - USB Mass Storage Device
  - USB Root Hub
  - USB Root Hub
  - USB Root Hub
  - USB Root Hub

Fig 1. Windows hardware device manager

## 4. References

- Philips LPC2141/2/4/6/8 User Manual, Philips Semiconductor, May, 2005
- Keil MCB2140 User Guide
- LPC2148 USB Mass Storage Device Example Code for Keil's MCB214x,  Feb. 6. 2006
- Universal Serial Bus Specification, Revision 2.0, The USB Implementers Forum (USB-IF), April. 2000

# 5. Legal information

## 5.1 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. Philips 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.

## 5.2 Disclaimers

**General —** Information in this document is believed to be accurate and reliable. However, Philips 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.

**Right to make changes —** Philips 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 —** Philips Semiconductors products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of a Philips Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. Philips Semiconductors accepts no liability for inclusion and/or use of Philips Semiconductors products in such equipment or applications and therefore such inclusion and/or use is for the customer's own risk.

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

## 5.3 Trademarks

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

## 6. Contents

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

For more information, please visit: http://www.semiconductors.philips.com
For sales office addresses, email to: sales.addresses@www.semiconductors.philips.com

**Date of release: 27 June 2006**

**Document identifier: AN10493_1**