AN10703

NXP USB host lite

Rev. 01 — 14 July 2008 Application note

Document information

Info	Content
Keywords	USB host lite, OHCI, LPC2468
Abstract	This application note illustrates how to access files on a USB mass storage device connected to the LPC2468 USB host port.

Revision history

Rev	Date	Description
01	20080714	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. Introduction

NXP Semiconductors LPC2400 family MCU provides an expandable, Plug and Play USB host serial interface, as seen on a typical PC environment that ensures a standard, low-cost connection for peripheral devices such as USB Device storage devices, keyboards, mouse, printers, scanners, and digital cameras.

The USB Host Controller allocates the USB bandwidth to attached devices through a token based protocol. It enables full- and low-speed data exchange with USB devices attached to the bus. The bus supports hot plugging, un-plugging, and dynamic configuration of the devices. All transactions are initiated by the host controller.

The USB Host Controller on LPC24xx consists of register interface, serial interface engine and DMA controller. The USB host register interface complies with the OHCI (Open Host Controller Interface) specification.

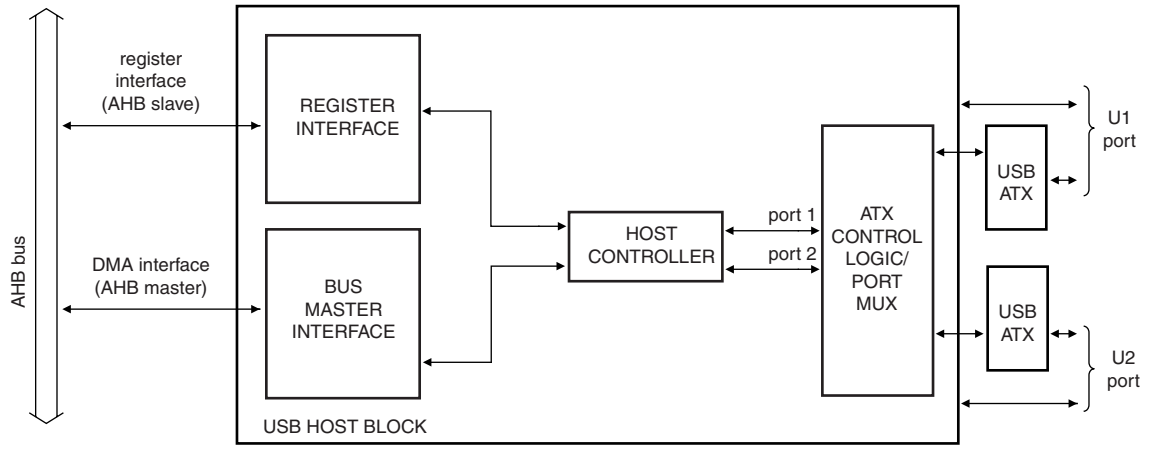

Fig 1. USB host controller block diagram

The main feature of USB Host Controller on LPC24xx includes:

- OpenHCI compliant.
- Supports all USB transfer types: Control, Bulk, interrupt and Isochronous.
- Contains integrated root hub and port manager.
- Contains four USB states visible to the SW Driver, USBOperational, USBReset, USBSuspend, USBResume.
- USB Frame management.
- Periodic listing and non-periodic listing process.
- Supports remote-wakeup
- Two downstream ports, both ports can be configured as host.
- Supports per-port power switching.

OnChip Technologies and NXP Semiconductors have partnered to provide USBHostLite software for LPC2468 micro controllers. USBHostLite is a stripped down stack that includes only USB mass storage class with the bare minimum code to run in an OS-less

environment. USBHostLite provides a simple solution for accessing the files on USB mass storage devices such as USB Pen Drives, USB Hard Disk Drives etc., connected to the LPC2468 USB Host port. USBHostLite source code is available from NXP website.

2. Specifications

This USBHostLite driver controls the LPC2468 embedded USB host controller hardware It is responsible for generating the appropriate USB transactions required to carry out USB requests issued by the protocol layer. The driver exposes a programming interface that supports USB setup requests and bulk transfers from/to individual endpoints. In addition, the driver supports USB device enumeration and address assignment.

The USBHostLite also includes a simple FAT16 file system driver. The file system driver maps logical objects to structures on the physical storage media. The file system driver exposes a software interface that provides standard file operations such as file create, open, close, read, write, etc.

The USBHostLite supports the following features:

1. Runs without any operating system.
2. Contains small memory foot print.
3. Source code is written in ANSI C and is independent of IDE.
4. Supports control and bulk transfers.
5. Supports FAT16 file system.
6. Simple File API allows file read and write operations.
7. Adheres to well defined coding standards by Onchip technologies.
8. Well structured and easily understandable source code.

3. Limitations

The complimentary USBHostLite from NXP has the following limitations:

1. USBHostLite supports only NXP LPC2468 micro controllers.
2. Classes other than the mass storage are not supported.
3. The mass storage interface must be present in the first configuration.
4. Max Logical Unit Number (LUN) greater than zero not supported.
5. File systems other than FAT16 are not supported.
6. Long filenames are not supported.
7. Files located in folders other than root directory can not be accessed.
8. The buffer size used in the application must not exceed 4kB.

4. Fully featured embedded USB stacks

For customer applications requiring full featured and production quality Embedded USB stacks, OnChip Technologies LLC provides Embedded USB Host/Device/OTG stacks that are fully compliant with the USB Specifications and offers great degree of stability and configurability.

5. Source code

The USBHostLite is contained in the UsbHost main directory. The subdirectories contain the additional sub components of USBHostLite.

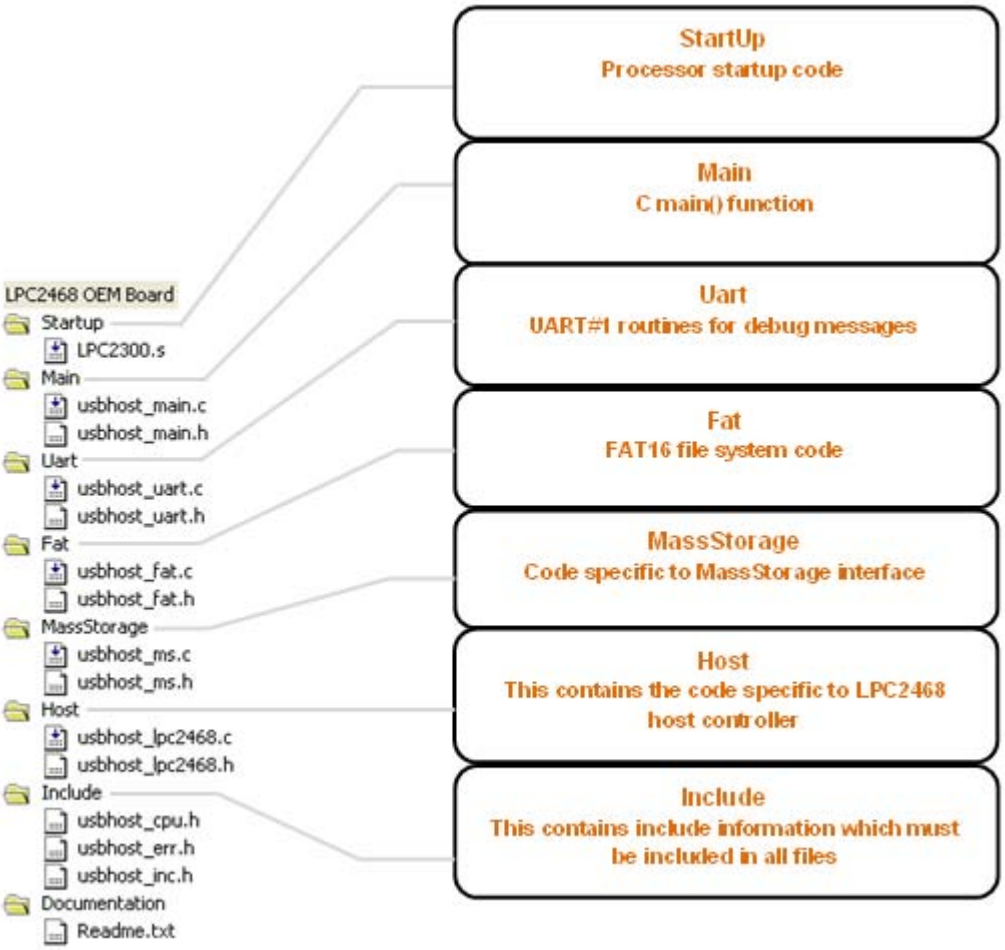

Fig 2. USBHostLite directory structure

Table 1. Functional description

Function	Description
Host_Init()	Initialize the host controller interface. This function allocates memory for all the global variables, enables the host clock, power up the LEDs and host ports, sets the frame interval, puts the host controller in operational state, enables the interrupts, and registers the interrupt service routine.
Host_EnumDev()	This function gets the device descriptor from the control endpoint i.e. endpoint 0 to determine the maximum packet size. Then it sets the device address to 1. It reads the first configuration and parses this configuration information to get the mass storage interface. If the mass storage interface is found, then it selects this configuration.
MS_Init()	Initialize the Mass storage interface. This function issues the necessary

Function	Description
	requests needed for any mass storage device to properly work. These requests are GetMaxLUN, TestUnitReady, GetSenseInfo and ReadCapacity. GetMaxLUN is the mass storage class specific request and all the other are SCSI commands.
FAT_Init()	Initialize the FAT16 file system interface. This function reads the boot sector from the USB mass storage device and stores the important information in a structure called BOOT_SEC. The file system uses this information for read/write operations from/to the USB device.
FILE_Open()	The file system uses this function to open a file that is located under the root directory. It searches for the given filename in the root directory. If the filename found, then it stores the attributes of that file in a structure FILE_ENTRY. If the file not found, the creation of the file depends on the flags given by the user. Flags = RDONLY Don't create the file Flags = RDWR Create the file if it doesn't exists.
FILE_Read()	Read the user requested bytes of data from the file given by the user into a buffer.
FILE_Write()	Write the user requested bytes of data from the buffer to the file given by the user.
FILE_Close()	Closes the file

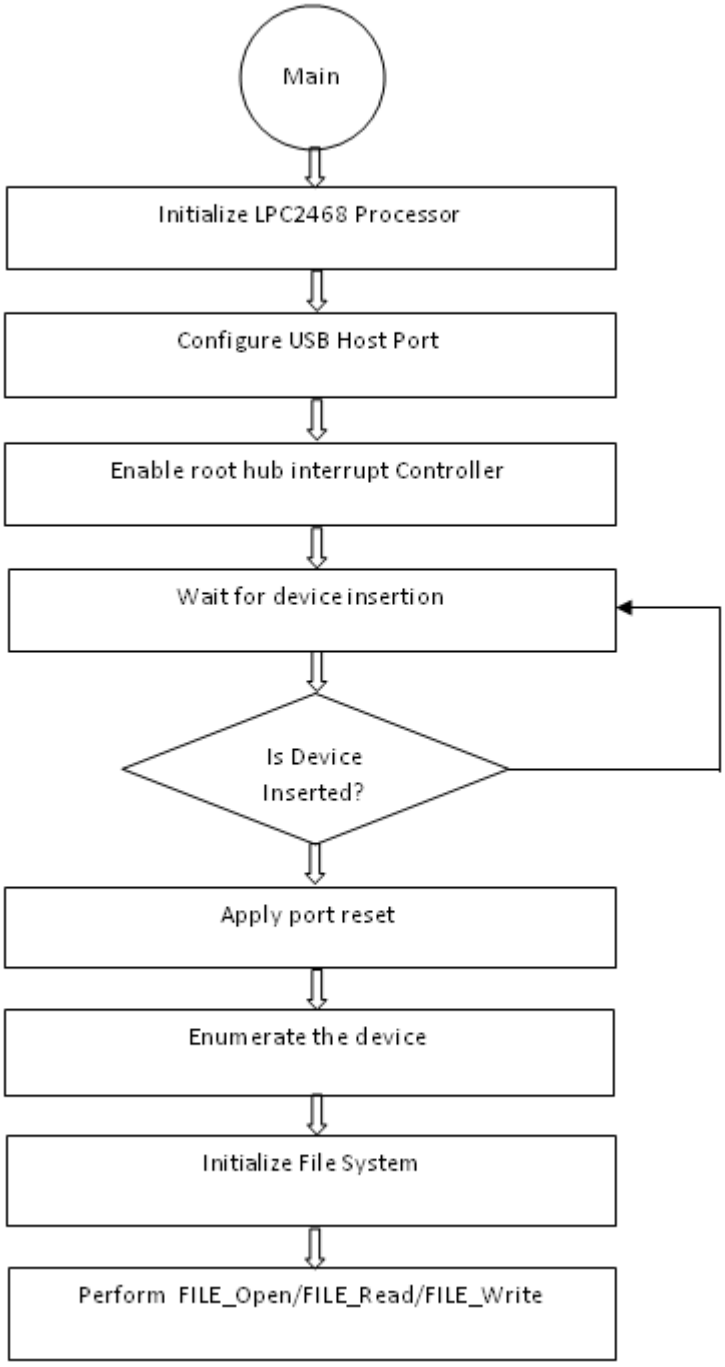

Fig 3. USBHostLite flowchart

6. Setup instructions

1. To run the mass storage sample, you need the following.

- Embedded Artists LPC2468 OEM board.
- Keil ULink USB-JTAG adapter for flash programming.
- Keil uVision3 IDE to open the project.
- Serial cable to see the log messages on the Hyper Terminal.

Jumper Settings of the LPC2468 board:

U2 as USB Host

P0.14 2-3

P1.30 2-3

USB-B+ 2-3

USB-B- 2-3

2. How to see the log messages on Hyper Terminal

Connect a serial cable to UART #1 on the LPC2468 board.

Configure the Hyper Terminal settings as 115200-N-8-1 and "No flow control".

Note: The following macros must be defined in the usbhost_lpc2468.h file.

#define PRINT_Log UART_Printf

#define PRINT_Err(rc) UART_Printf("ERROR: In %s at Line %u - rc = %d\n", __FUNCTION__, __LINE__, rc)

3. The user is provided with three sample functions, Main_Read(), Main_Write() and Main_Copy() defined in usbhost_main.c.

If the user calls Main_Read(), the host sample reads data from file "FILENAME_R" into a buffer.

If the user calls Main_Write(), the host sample writes data to file "FILENAME_W" from a buffer.

If the user calls Main_Copy(), the host sample copies data from file "FILENAME_R" to file "FILENAME_W".

Note: The file name defined by macro "FILENAME_R" must be present on the device (usb flash drive).

4. How to download and run the demo:

i. Connect the power supply by using USB or optional power supply.

ii. Connect the Keil U-Link USB J-Tag

iii. Connect the serial cable at UART#1 to see the log messages.

iv. Open the hyper terminal and make settings as shown in Hyper terminal settings.

v. Open the SampleUsbHost directory.

vi. Open the project SampleUsbHost.Uv2 by double clicking on it.

vii. Goto project tab and click 'Rebuild all target files' to compile the project.

viii. Goto Flash tab and click 'Download'.

ix. Connect a USB flash drive at USB host port.

Note: Make sure that all the jumper settings are as shown in Fig 4.

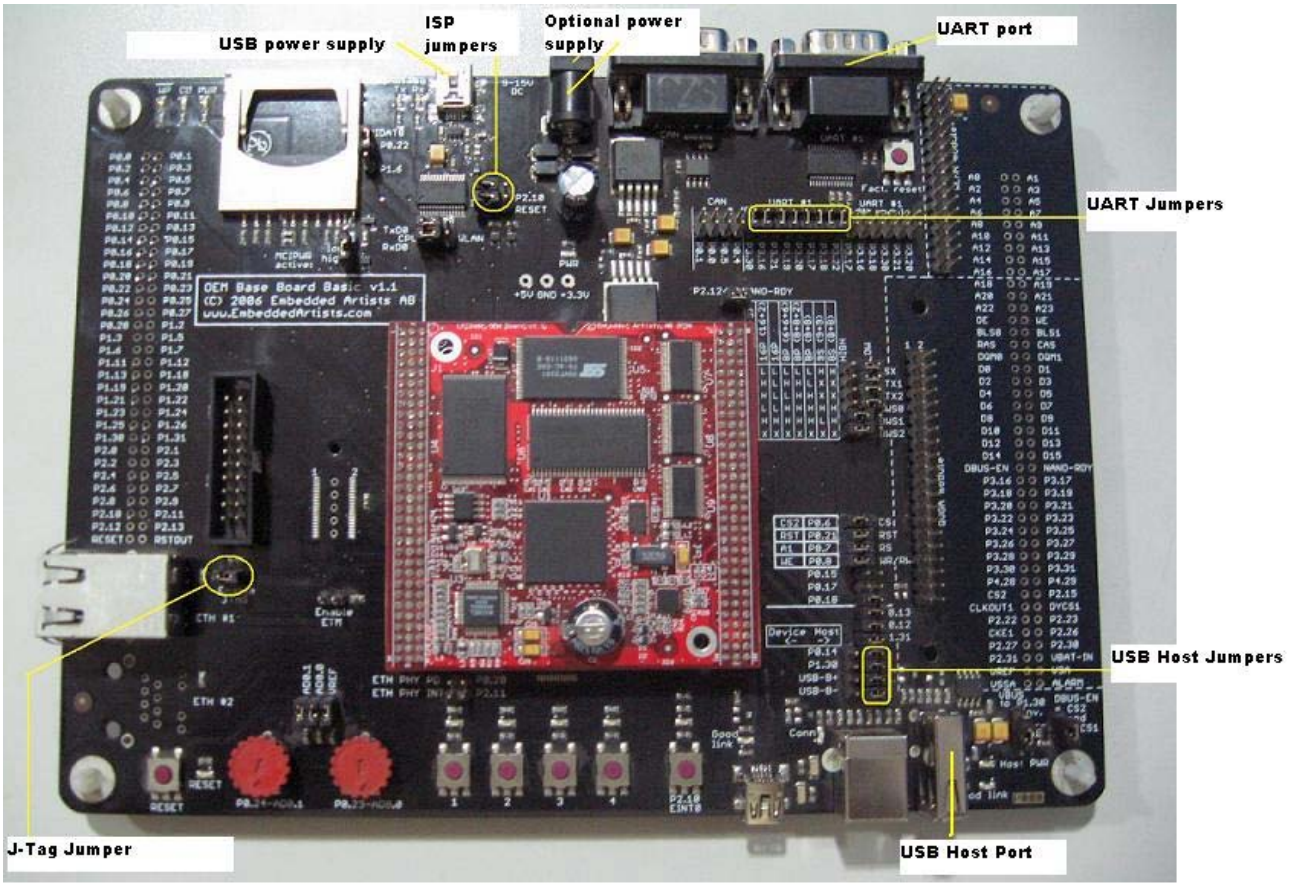

Fig 4. Embedded Artists LPC2468 OEM board

7. References

- NXP LPC2468 User Manual (UM10237) http://www.standardics.nxp.com/support/documents/?search=UM10237, NXP Semiconductors.
- Universal Serial Bus Specification, Revision 2.0, The USB Implementers Forum (USB-IF), April, 2000.
- Open Host Controller Interface Specification for USB, rev. 1.0a, September, 1999.

8. About NXP

NXP creates semiconductors, system solutions and software that deliver better sensory experiences in mobile phones, personal media players, TVs, set-top boxes, identification applications, cars and a wide range of other electronic devices.

http://www.standardics.nxp.com/microcontrollers/

9. About OnChip Technologies

OnChip Technologies LLC is a leading supplier of ready to use embedded software components. OnChip is highly committed to deliver high quality software with clean source code, documentation and support. OnChip's products and software design services help companies bring up their products quickly and cost effectively.

http://www.nxp.com/redirect/onchiptech.com

10. Legal information

10.1 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.

10.2 Disclaimers

General — 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.

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 medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of a NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP 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. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

10.3 Trademarks

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

11. Contents

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

Date of release: 14 July 2008
Document identifier: AN10703_1