AN11018

USB composite device on the LPC134x

Rev. 1 — 17 December 2010 Application note

Document information

Info	Content
Keywords	USB, virtual COM, mass storage, composite, Microcontroller, descriptor, Interface Association Descriptor, IAD, CDC, ACM, MSD
Abstract	This document describes how to create a composite USB device with an interface association descriptor for the LPC134x USB port.

Revision history

Rev	Date	Description
1	20101217	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

This is an example of a composite device driver for the USB device block on the LPX134x microcontrollers. A composite device is one that has multiple interfaces controlled independently of each other. In this device the independent interfaces include:

- virtual COM device (CDC/ACM)
- mass storage device

2. Driver architecture

This driver was created by combining the code in the usbcdc (virtual COM port) and usbmsd (mass storage) example projects of the LPCXpresso IDE. As these two examples were already very similar to each other the merge of the functional code included few changes, the bulk of changes were made to the descriptors.

2.1 Functional code

The code in this example driver is very simple. It includes a main() function with an endless loop and a collection of library support routines that implement the basic USB device stack and provide support for the CDC/ADM and MSD device classes.

The main() function performs a onetime initialization of the USB stack and classes and then drops into a loop that processes characters over the communications channel. All mass storage and character TX/RX activity is interrupt driven.

The mass storage read and write handlers operate on a 4K RAM buffer only, the device's internal flash is not used in this example.

2.2 Descriptors

During the enumeration process a USB device will at some point send a collection of descriptors to the host that provide detailed information about itself, such as what class or classes the device supports and how it will interface to the host.

This section will discuss those specific areas in the descriptors that make this a composite device. Reference the USB 2.0 specification at usb.org for a full description of the descriptors.

2.2.1 Descriptor types and ordering

This device produces a device descriptor with a Miscellaneous Device Class code, a single configuration descriptor, a single Interface Association Descriptor (IAD), and three interface descriptors.

The ordering of these descriptors in memory is important. They are ordered like this:

1. Device descriptor
2. Configuration descriptor
3. Interface descriptor (MSD)
4. Interface Association Descriptor (CDC)
5. Interface Descriptor (CDC communications class)
6. Interface Descriptor (CDC data class)

2.2.2 Device descriptor

The device descriptor describes general information about the device. The key fields in this descriptor that define this device as composite are the device class, subclass, and protocol. These three fields are not filled with zeros as is usually the case with composite devices because the configuration descriptor contains an IAD descriptor.


```
/* USB Standard Device Descriptor */
const uint8_t USB_DeviceDescriptor[] = {
  USB_DEVICE_DESC_SIZE,               /* bLength */
  USB_DEVICE_DESCRIPTOR_TYPE,         /* bDescriptorType */
  WBVAL(0x0200), /* 2.0 */            /* bcdUSB */
  USB_DEVICE_CLASS_MISCELLANEOUS,     /* bDeviceClass */
  0x02,                               /* bDeviceSubClass */
  0x01,                               /* bDeviceProtocol */
  USB_MAX_PACKET0,                    /* bMaxPacketSize0 */
  WBVAL(USB_VENDOR_ID),               /* idVendor */
  WBVAL(USB_PROD_ID),                 /* idProduct */
  WBVAL(USB_DEVICE), /* 1.00 */       /* bcdDevice */
  0x01,                               /* iManufacturer */
  0x02,                               /* iProduct */
  0x03,                               /* iSerialNumber */
  0x01                                /* bNumConfigurations: one
possible configuration*/
};
```

NOTE: The constant USB_DEVICE_CLASS_MISCELLANEOUS is equal to 0xEF.

2.2.3 Interface association descriptor

The Interface Association Descriptor associates multiple interfaces with a single logical function. In this device the multiple interfaces are the two CDC interfaces, communications and data, and the single logical function is the CDC device class.

This device includes one of these descriptors because the CDC interfaces are part of a composite device class that includes other interfaces. Had this device not been a composite device then the device class code defined in the device descriptor would have been set to USB_DEVICE_CLASS_COMMUNICATIONS, which automatically makes the association, and there would have been no need to use this descriptor. Without the IAD the OS would attempt to load a separate driver for each of the CDC interfaces.

```
  /* IAD to associate the two CDC interfaces */
  USB_INTERFACE_ASSOCIATION_DESC_SIZE,       /* bLength */
  USB_INTERFACE_ASSOCIATION_DESCRIPTOR_TYPE, /* bDescriptorType */
  USB_CDC_CIF_NUM,                   /* bFirstInterface */
  2,                                 /* bInterfaceCount */
  USB_DEVICE_CLASS_COMMUNICATIONS,   /* bFunctionClass */
  CDC_ABSTRACT_CONTROL_MODEL,        /* bFunctionSubClass */
  0,                                 /* bFunctionProtocol */
  0,                                 /* iFunction (Index of string
descriptor describing this function) */
```

See http://www.usb.org/developers/whitepapers/iadclasscode_r10.pdf for more details about IADs.

This IAD associates the 2 interfaces by including the number of the first interface to be associated in the bFirstInterface field and the count of interfaces to be associated together in the bInterfaceCount field. In this case there are two interfaces that are to be associated with each other and they start with the one numbered `USB_CDC_CIF_NUM`.

NOTE: only contiguously numbered interfaces can be associated. Also note that this IAD descriptor must be positioned just before the interfaces it references.

3. Windows OS support

The file lpc134x-vcom.inf exists to assist in the loading of the proper USB serial drivers in the Windows operating system for this device to operate correctly as a virtual COM interface.

There is a line in this file titled [DeviceList] which looks like this:

```
%DESCRIPTION%=LPC134xUSB, USB\VID_1FC9&PID_0003&MI_01
```

This line associates the CDC device with the operating system's usbser.sys driver file and causes the OS to load this driver during the enumeration process.

VID_1FC9 and PID_0003 are the vendor and product IDs for this device. The MI_01 portion of the line (MI stands for Multiple Interface) corresponds to the second interface in the device and tells the OS to associate this interface with the driver that is loaded. The CDC device is the second interface because it was the second interface descriptor defined after the configuration descriptor as shown in 2.2.1 Descriptors.

Since this composite device includes a mass storage device, and since this mass storage device is enumerated before the CDC device, this .inf file can be stored in the device and accessed directly by the operating system when the CDC device is being enumerated.

4. Legal information

4.1 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.

4.2 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.

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 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 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 national 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.

4.3 Trademarks

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

5. 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: 17 December 2010
Document identifier: AN11018