Future Technology Devices International Ltd.

Technical Note TN_113

Simplified Description of USB Device Enumeration

Document Reference No.: FT_000180

Version 1.0

Issue Date: 2009-10-28

USB Enumeration is the process of detecting, identifying and loading drivers for a USB device.

The purpose of this document is to provide an overview of the mechanics of the process.

Future Technology Devices International Limited (FTDI)

Unit1, 2 Seaward Place, Centurion Business Park, Glasgow G41 1HH United Kingdom
Tel.: +44 (0) 141 429 2777   Fax: + 44 (0) 141 429 2758
E-Mail (Support): **support1@ftdichip.com** Web: http://www.ftdichip.com

TABLE OF CONTENTS

1 Introduction

USB Enumeration is the process of detecting, identifying and loading drivers for a USB device.

This involves a mixture of hardware techniques for detecting something is present and software to identify what has been connected.

The purpose of this document is to provide an overview of the mechanics of the process.

2 Detecting a Device has been Connected

A USB interface consists of 4 wires. Power, Ground, Data Plus (USBDP) and Data Minus (USBDM). A USB host port with no devices connected uses 15kohm resistors to connect both USB DP and USB DM to GND.

When a USB device (sometimes referred to as a slave) is plugged into a USB host there is a change on these USB data lines. It is this change that the USB host uses to detect a device has been connected. This change is also used to identify the speed of device attached.

3 Determining the Device Speed

A low speed USB device (1.5Mbps) uses a 1k5 pull-up resistor to VCC on the USB DM signal line.

A full speed USB device (12Mbps) uses a 1k5 pull-up resistor to VCC on the USB DP signal line.

A high speed USB device (480Mbps) will initially appear as a full speed device to the host. The first thing the USB host does is to attempt to send /receive packets at high speed to the USB device. This is known as J and K chirp and if communication is successful it will be assumed that the USB device is a high speed device. If this initial communication fails then the USB host assumes that the device is a full speed device.
This means a high speed device has a 1k5 pull up resistor on USB DP that can be switched in / out of circuit.

A J state is defined as a differential signal on USBDP and USB DM >= +300mV.
A K state is defined as a differential signal on USBDP and USB DM >= -300mV.

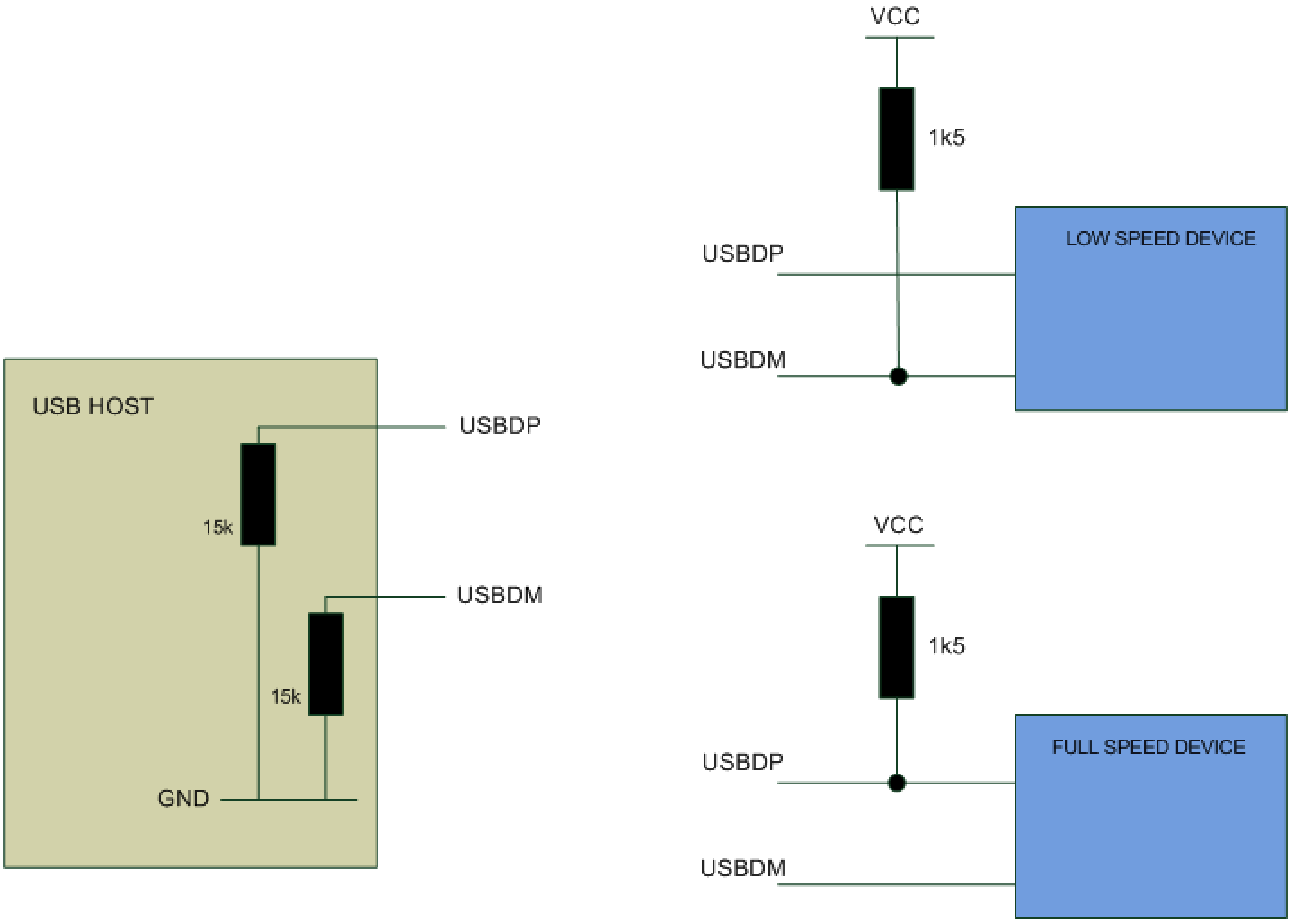


Figure 1: Connection of low / full speed devices.

4 Determining What Device is Attached (Device Descriptor)

Devices are identified by descriptors.

Once the USB host has established a USB device is connected, and at what speed it should communicate, then the host will reset the USB device and attempt to read the descriptors to identify the USB device using a default address.

This basically follows a question and answer process. The USB host will send a Get_Device_Descriptor command and then receive a packet of bytes with the descriptor length and the actual descriptor.

At the completion of this stage the device is reset and given a unique address before getting the configuration and interface descriptors.

bLength = 18
bDescriptorType = 1
bcdUSB
bDeviceClass
bDeviceSubClass
bDeviceProtocol
bMaxPacketSize
idVendor
idProduct
bcdDevice
iManufacturer
iProduct
iSerialNumber
bNumConfigurations

4.1 bLength.

All USB devices have descriptors and the first key one is the Device Descriptor. The length is 18 bytes and it is bDdescriptorType is type 1.

4.2 bcdUSB.

This is used to identify the device as a USB 1.0, USB 1.1 or USB 2.0 compliant device.

4.3 bDeviceClass, bDeviceSubClass and bProtocol

The bDeviceClass of device defines the device type e.g. a USB Mouse is a Human Interface Device (HID) class device. This is given a hex value of 0x03.

More complex devices such as Communication Device Class (CDC) may also use a sub class to break down the device type into a smaller group.

bProtocol is used to qualify the sub class.

For a full list of defined device classes see Appendix B.

4.4 bMaxPacketSize

This defines the maximum number of bytes in a packet from an endpoint

4.5 idVendor and idProduct

The idVendor (VID) is assigned to a company by the USB Implementers Forum. An idProduct (PID) is used with this value to help associate a device with a manufacturer and product. It is also used to help link the hardware with a specific driver. (see AN_107 Advanced Driver Options for more information on driver inf files).

4.6 iManufacturer, iProduct and iSerialNumber

These values are indexes to the Manufacturer string, the Product name string, and the Serial Number strings. These descriptors help make the identifiers more human readable and can be of variable length.

5 Determining the Device Configuration (Configuration Descriptor)

The Configuration Descriptor is of a fixed length – 9 bytes and is defined as type 2. This provides device specific information such as the number of interfaces supported by the device and maximum power the device is expected to consume.

bLength = 9
bDescriptorType = 2
wTotallength
bNumInterfaces
bConfigurationValue
iConfiguration
bmAttributes
bMaxPower

5.1 wTotaLength

This parameter defines the total length of the descriptor.

5.2 bNumInterfaces

This parameter defines the number of interfaces supported

5.3 bConfigurationValue

Value to use as an argument to the SetConfiguration() request to select this configuration

5.4 iConfiguration

This parameter gives the index of the strings to describe the configuration.

5.5 bmAttributes

This parameter defines specific configuration attributes e.g. support for remote wake or whether the device is bus [powered or self powered.

5.6 bMaxPower

This parameter defines the maximum power the device is expected to consume.

6 Determining the Device Interface (Interface Descriptor)

The Interface Descriptor is of a fixed length – 9 bytes and is defined as type 4. This provides information such as the number of endpoints to be used e.g. FTDI devices use one IN and one OUT for each interface.

bLength = 9
bDescriptorType = 4
bInterfaceNumber
bAlternateSetting
bNumEndpoints
bInterfaceClass
bInterfaceSubClass
bInterfaceProtocol
iInterface

6.1 bInterfaceNumber

This parameter defines the number of the interface being described.

6.2 bAlternateSetting

This parameter is used to select this alternate setting for the interface selected by bInterfaceNumber.

6.3 bNumEndpoints

This parameter defines the number of endpoints used by this interface.

6.4 bInterfaceClass, bInterfaceSubClass and bInterfaceProtocol

These parameters define the class of the interface similar to the classes defined in the device descriptor.

6.5 iInterface

This parameter provides the index for the string descriptor used to describe this interface.

7 Loading the Driver

When the USB device has been fully identified by the USB host, then the host PC needs a driver to control the USB device. Matching the USB device to the driver is usually done by matching up the VID and PID.

With an FTDI USB device, the VID/PID are stored in the driver .inf files (Windows OS). If the VID/PID match, the host PC will know which driver to install for the specific device. (see AN_107 Advanced Driver Options for more information on driver inf files).

After the initial installation the settings are saved in the PC registry so that on subsequent plug-ins of the USB device, the driver is automatically loaded.

When the driver has been loaded, then the USB device is available for applications to access.

8 Summary

Enumeration of a device is not a trivial matter. The enumeration processes described in this FTDI technical note are provided as background reading for engineers looking for a little bit extra understanding of the USB protocol.

When using an FTDI device, the enumeration process is hidden from the end user by FTDI silicon and free FTDI drivers. This simplifies USB interface design, when compared to a software solution, by removing a large part of the complexity of USB design and allows designers to focus on the interface and application design beyond the USB port.

9 Contact Information

Head Office – Glasgow, UK

Future Technology Devices International Limited
Unit 1, 2 Seaward Place, Centurion Business Park
Glasgow G41 1HH
United Kingdom

Tel: +44 (0) 141 429 2777
Fax: +44 (0) 141 429 2758

E-mail (Sales) sales1@ftdichip.com
E-mail (Support) support1@ftdichip.com
E-mail (General Enquiries) admin1@ftdichip.com
Web Site URL http://www.ftdichip.com
Web Shop URL http://www.ftdichip.com

Branch Office – Taipei, Taiwan

Future Technology Devices International Limited (Taiwan)
2F, No 516, Sec. 1 NeiHu Road
Taipei 114
Taiwan, R.O.C.
Tel: +886 (0) 2 8797 1330
Fax: +886 (0) 2 8751 9737

E-mail (Sales) tw.sales1@ftdichip.com
E-mail (Support) tw.support1@ftdichip.com
E-mail (General Enquiries) tw.admin1@ftdichip.com
Web Site URL http://www.ftdichip.com

Branch Office – Hillsboro, Oregon, USA

Future Technology Devices International Limited (USA)
7235 NW Evergreen Parkway, Suite 600
Hillsboro, OR 97123-5803
USA
Tel: +1 (503) 547 0988
Fax: +1 (503) 547 0987

E-Mail (Sales) us.sales@ftdichip.com
E-Mail (Support) us.admin@ftdichip.com
Web Site URL http://www.ftdichip.com

Branch Office – Shanghai, China

Future Technology Devices International Limited (China)
Room 408, 317 Xianxia Road,
ChangNing District,
ShangHai, China

Tel: +86 (21) 62351596
Fax: +86(21) 62351595

E-Mail (Sales): cn.sales@ftdichip.com
E-Mail (Support): cn.support@ftdichip.com
E-Mail (General Enquiries): cn.admin1@ftdichip.com
Web Site URL: http://www.ftdichip.com

Distributor and Sales Representatives

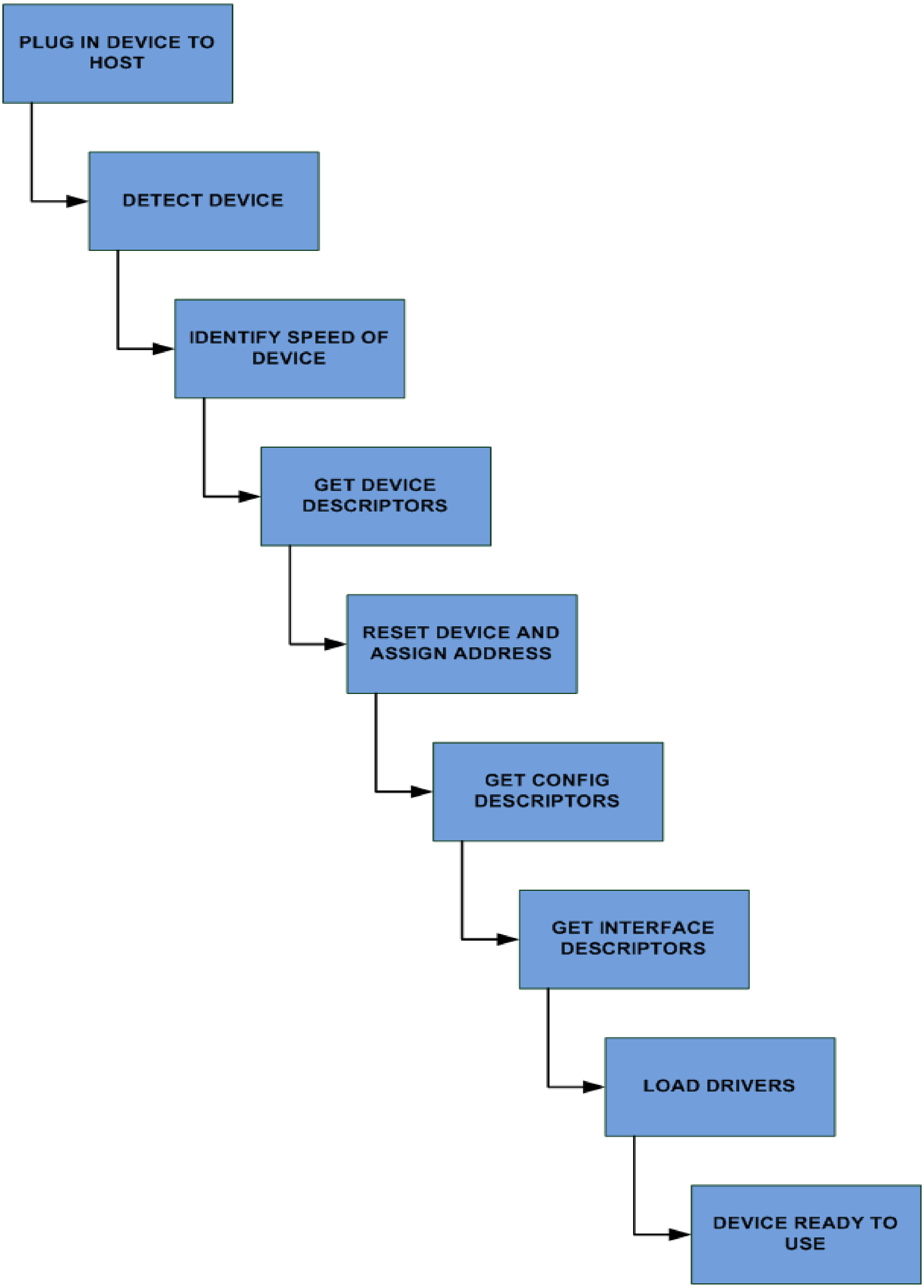
Please visit the Sales Network page of the FTDI Web site for the contact details of our distributor(s) and sales representative(s) in your country.

Neither the whole nor any part of the information contained in, or the product described in this manual, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder. This product and its documentation are supplied on an as-is basis and no warranty as to their suitability for any particular purpose is either made or implied. Future Technology Devices International Ltd will not accept any claim for damages howsoever arising as a result of use or failure of this product. Your statutory rights are not affected. This product or any variant of it is not intended for use in any medical appliance, device or system in which the failure of the product might reasonably be expected to result in personal injury. This document provides preliminary information that may be subject to change without notice. No freedom to use patents or other intellectual property rights is implied by the publication of this document. Future Technology Devices International Ltd, Unit 1, 2 Seaward Place, Centurion Business Park, Glasgow G41 1HH United Kingdom. Scotland Registered Number: SC136640

Appendix A – Terminology

FTDI	Future Technology Devices International
USB	Universal Serial Bus
USBDP	USB Data line plus
USBDM	USB Data line minus
USB Host	The port that controls the USB interface – the master.
USB Device	The peripheral that connects to the USB host port – the slave.
VID	idVendor
PID	idProduct

Appendix B – Enumeration Flow



Appendix C – Device Classes

Class (hex value)	Definition
00	USE Class information in the Interface Descriptors
01	Audio
02	Communications and CDC Control
03	HID
05	Physical
06	Image
07	Printer
09	Mass Storage
09	Hub
0A	CDC-Data
0B	Smart Card
0D	Content Security
0E	Video
0F	Personal Healthcare
DC	Diagnostic Device
E0	Wireless Controller
EF	Miscellaneous
FE	Application Specific
FF	Vendor Specific

More information on device classes can be found at www.USB.org

Appendix D – Standard Descriptor Types

Descriptor Type	Descriptor Name
1	Device
2	Configuration
3	String
4	Interface
5	Endpoint
6	Device Qualifier
7	Other Speed Configuration
8	Interface Power

More information on device classes can be found at www.USB.org

Appendix E – References

USB Implementers Forum – www.usb.org

USB Specification - http://www.usb.org/developers/docs/usb_20_052709.zip

AN_107 Advanced Driver options.
http://www.ftdichip.com/Documents/AppNotes/AN_107_AdvancedDriverOptions_AN_000073.pdf

Appendix F – Revision History

Version 1.0	First Release	28/10/2009