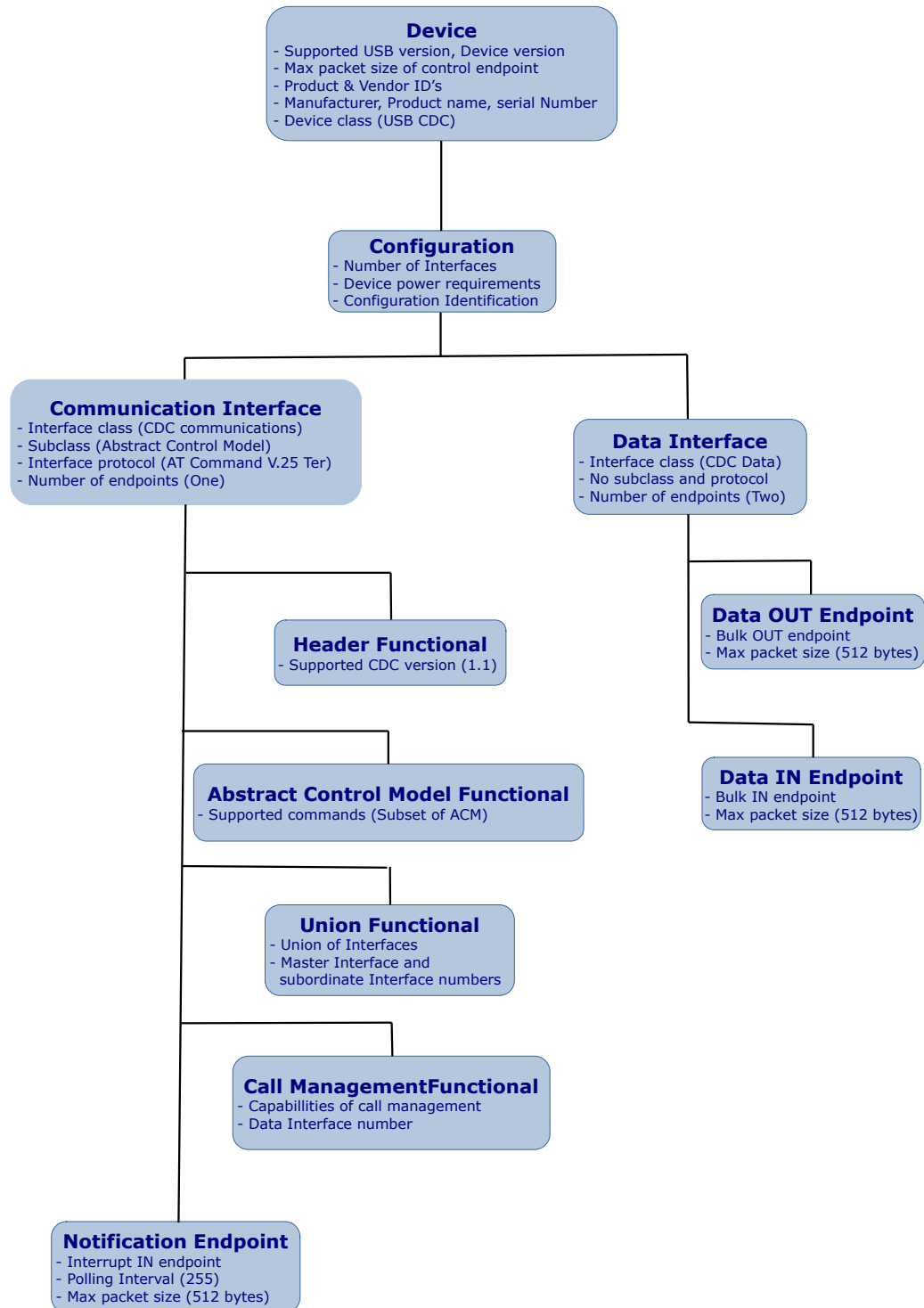


Configuration fields for CDC communication

The diagram is a visual example of a configuration for one CDC device.

The description of the configuration fields is for a system with two CDC devices.



More information can be found on these websites:

www.usb.org/developers/docs/devclass_docs/

www.usb.org

www.keil.com/pack/doc/mw/usb/html

<https://beyondlogic.org/usbnutshell/usb1.shtml>

<https://developerhelp.microchip.com/xwiki/bin/view/applications/usb/how-it-works/>

<https://onlinedocs.microchip.com/oxy/GUID-38BB0804-1305-4C34-99DC-04025E667766-en-US-5/GUID-56B2B02B-BB4F-41F1-8774-BCA3B46CD210.html>

Dual CDC configuration descriptor set

The device descriptor of a USB device represents the entire device. As a result a USB device can only have one device descriptor. It specifies some basic, yet important information about the device such as the supported USB version, maximum packet size, vendor and product IDs and the number of possible configurations the device can have.

Device descriptor		
bLength	0x12	Size of the descriptor in byte (18 bytes)
bDescriptorType	0x01	Descriptor type (0x01 = device)
bcdUSB	0x0110	USB version number (BCD) 1.10
bDeviceClass	0xEF	Class code (assigned by USB-IF) 0x00 specified by each interface 0xFF specified by vendor otherwise a valid class code: 0xEF miscellaneous device class code
bDeviceSubClass	0x02	Subclass code (assigned by USB-IF) 0x02 Communication device subclass code, miscellaneous
bDeviceProtocol	0x01	Protocol code (assigned by USB-IF) 0x01 Communication device protocol code, miscellaneous
bMaxPacketSize0	0x40	Maximum packet size for endpoint zero (8, 16, 32 or 64)
idVendor	0x1209	Vendor ID (assigned by USB-IF) 0x6666 Prototype product vendor ID 0x1209 Generic pid.codes for open source hardware projects
idProduct	0xB126	Product ID (assigned by USB-IF) 0x6610 unknown i.c.w. vendor ID 0x6666 0xB126 unknown i.c.w. vendor ID 0x1209
bcdDevice	0x0100	Device version number 1.00
iManufacturer	0x00	Index of manufacturer string descriptor
iProduct	0x02	Index of product string descriptor
iSerialNumber	0x00	Index of serial number string descriptor
bNumConfigurations	0x01	Number of supported configurations

[Back to top](#)

A USB device can have several different configurations although the majority of devices are simple and only have one. The configuration descriptor specifies how the device is powered, what the maximum power consumption is, the number of interfaces it has. Therefore it is possible to have two configurations, one for when the device is bus powered and another when it is mains powered. As this is a "header" to the Interface descriptors, its also feasible to have one configuration using a different transfer mode to that of another configuration.

Configuration descriptor		
bLength	0x09	Size of the descriptor in byte (9 bytes)
bDescriptorType	0x02	Descriptor type (0x02 = configuration)
wTotalLength	0x008D	The number of bytes in the configuration descriptor and its subordinate descriptors
bNumInterfaces	0x04	Number of interfaces in the configuration (4 data endpoints)
bConfigurationValue	0x01	Value that identifies the configuration selected
iConfiguration	0x00	Index of string descriptor for configuration details
bmAttributes	0x80	Self/bus powered and wakeup settings (80 = Bus powered)
bMaxPower	0xFA	Device current requirements (250 x 2mA = 500mA)

[Back to top](#)

CDC 0

The interface descriptor could be seen as a header or grouping of the endpoints into a functional group performing a single feature of the device.

Interface Association Descriptor		
bLength	0x08	Size of the descriptor in bytes (8 bytes)
bDescriptorType	0x0B	Descriptor type (0x0B = interface association)
bFirstInterface	0x00	First associated interface number of the set of interfaces that follow this descriptor.
bInterfaceCount	0x02	The Number of interfaces follow this descriptor that are considered "associated".
bFunctionClass	0x02	Class code (assigned by USB-IF) 0x02 = communications and CDC control See table at end of document
bFunctionSubClass	0x02	Subclass code (assigned by USB-IF) See table at end of document
bFunctionProtocol	0x00	Protocol code (assigned by USB-IF) See table at end of document
iFunction	0x00	Index of string descriptor describing the associated interfaces.

[Back to top](#)

The USB Interface Control Descriptor is a data structure defined in the USB specification that describes the characteristics of a specific interface within a USB device configuration.

Interface 1 Control Descriptor		
bLength	0x09	Size of the descriptor in bytes
bDescriptorType	0x04	Descriptor type (0x04 = interface)
bInterfaceNumber	0x00	Identify number for this interface
bAlternateSetting	0x00	Selected alternate setting for interface
bNumEndpoints	0x01	Number of endpoints supported by interface
bInterfaceClass	0x02	Class code (assigned by USB-IF) 0x02 = communications and CDC control See table at end of document
bInterfaceSubClass	0x02	Subclass code (assigned by USB-IF) See table at end of document
bInterfaceProtocol	0x00	Protocol code (assigned by USB-IF) See table at end of document
iInterface	0x04	Index of string descriptor describing interface

[Back to top](#)

The USB Header Management Functional Descriptor is used in CDC devices (like virtual COM ports, modems, Ethernet adapters, etc.) to declare compliance with a specific version of the CDC specification and to act as the first functional descriptor in a set.

Header Management Functional Descriptor		
bFunctionalLength	0x05	Size of the descriptor in bytes
bDescriptorType	0x24	Descriptor type (0x24 = CS_INTERFACE)
bDescriptorSubtype	0x00	Subtype code (0x00 = Header Functional Descriptor, which marks the beginning of the concatenated set of functional descriptors for the interface)
bcdCDC	0x0110	USB Class Definitions for Communications Devices Specification release number (BCD)

[Back to top](#)

The Call Management Functional Descriptor tells the host how the device handles call management (telephone-style or modem-style communication) and which interface is used for call management.

Call Management Functional Descriptor		
bFunctionalLength	0x05	Size of the descriptor in bytes
bDescriptorType	0x24	Descriptor type (0x24 = CS_INTERFACE)
bDescriptorSubtype	0x01	Subtype code (0x01 = Call Management Functional Descriptor)
bmCapabilities	0x00	Bit 0: Device handles call management itself. Bit 1: Device can send/receive call management information over a data class Bit 2-7: Reserved
bDataInterface	0x01	Interface number of Data Class interface. Optionally used for call management

[Back to top](#)

The ACM Functional Descriptor specifically describes the capabilities of a device to support certain communication functionalities such as control, monitoring, and managing serial communication over USB.

Abstract Control Management Functional Descriptor.		
bFunctionalLength	0x04	Size of the descriptor in bytes
bDescriptorType	0x24	Descriptor type (0x24 = CS_INTERFACE)
bDescriptorSubtype	0x02	Subtype code (0x02 = Abstract Control Management Functional Descriptor)
bmCapabilities	0x06	Bit 0: Supports line coding and control line states. Bit 1: Supports break signal management. Bit 2: Supports network connection management.

[Back to top](#)

The Union Functional Descriptor is used in CDC devices to declare a relationship between one Communication Class Interface (the "master") and one or more Data Class Interfaces (the "slaves").

CDC Union Functional Descriptor		
bFunctionalLength	0x05	Size of the descriptor in bytes
bDescriptorType	0x24	Descriptor type (0x24 = CS_INTERFACE)
bDescriptorSubtype	0x06	Subclass code (0x06 = Union Functional Descriptor)
bMasterInterface	0x00	Interface number of the Communication class interface
bSlaveInterface0	0x01	First associated data class Interface
bSlaveInterfaceN	0x??	Additional slave interfaces (if any)

[Back to top](#)

Endpoint descriptors are used to describe endpoints other than endpoint zero. Endpoint zero is a control endpoint and is configured before any descriptors are requested. The host will use the information returned from these descriptors to determine the bandwidth requirements of the bus.

Endpoint 1 Interrupt Descriptor		
bLength	0x07	Size of the descriptor in bytes
bDescriptorType	0x05	Descriptor type (0x05 = endpoint)
bEndpointAddress	0x81	The address and direction of the endpoint
bmAttributes	0x03	Type of transfer: 2 = Bulk, 3 = Control Synchronisation and usage type for isochronous endpoint
wMaxPacketSize	0x0008	Maximum packet size 08 to 64 byte Additional transactions for high speed interrupt and isochronous endpoints
bInterval	0x10	Polling interval for interrupt and isochronous endpoints Maximum NAK rate for bulk and control OUT endpoints

[Back to top](#)

A Data Interface Descriptor is just a standard USB Interface Descriptor (not class-specific) that describes the Data Class interface belonging to a CDC device.

Data Interface 1 Descriptor		
bLength	0x09	Size of the descriptor in bytes
bDescriptorType	0x04	Descriptor type (0x04 = interface)
bInterfaceNumber	0x01	The number of this interface. Zero based value identifying the index in the array of concurrent interfaces supported by this configuration.
bAlternateSetting	0x00	Value used to select an alternate setting for the interface identified in the prior field. Allows an interface to change the settings on the fly.
bNumEndpoints	0x02	Number of endpoints used by this interface (excluding endpoint 0)
bInterfaceClass	0x0A	Class code (assigned by USB-IF) 0x0A CDC data class See table at end of document
bInterfaceSubClass	0x00	Subclass code (assigned by USB-IF) See table at end of document
bInterfaceProtocol	0x00	Protocol code (assigned by USB-IF) See table at end of document
iInterface	0x00	Index of string descriptors describing the interface

[Back to top](#)

An Endpoint Descriptor defines the characteristics of a specific endpoint within an interface of a USB device.

Endpoint 2 IN Descriptor		
bLength	0x07	Size of the descriptor in bytes
bDescriptorType	0x05	Descriptor type (0x05 = endpoint)
bEndpointAddress	0x82	bit 0..3 endpoint number bit 4..6 reserved. Set to 0 bit 7 direction 0=out 1=in
bmAttributes	0x02	bit 0..1 Transfer type 00=control 01=isochronous 10=bulk 11=interrupt bit 2..7 reserved. If isochronous endpoint bit 3..2 synchronisation type (iso mode) 00=no synchronisation 01=asynchronous 10=adaptive 11=synchronous bit 5..4 usage type (iso mode) 00=data endpoint 01=feedback endpoint 10=explicit feedback data endpoint 11=reserved Bit 7..6 Reserver
wMaxPacketSize	0x0040	Maximum packet size this endpoint is capable of sending or receiving
bInterval	0x00	Interval for polling endpoint data transfers. Value in frame counts. Ignored for bulk en control endpoints. Isochronous must equal 1 and field may range from 1 to 255 for interrupt endpoints

[Back to top](#)

An Endpoint Descriptor defines the characteristics of a specific endpoint within an interface of a USB device.
(See [endpoint 2 IN](#) for field info)

Endpoint 3 OUT Descriptor		
bLength	0x07	Size of the descriptor in bytes
bDescriptorType	0x05	Descriptor type (0x05 = endpoint)
bEndpointAddress	0x03	
bmAttributes	0x02	
wMaxPacketSize	0x0040	
bInterval	0x00	

[Back to top](#)

CDC 1

(This is a copy of CDC 0, using different endpoints)

Interface associate descriptor		
bLength	0x08	Size of the descriptor in byte (8 bytes)
bDescriptorType	0x0B	Descriptor type (0x0B = interface association)
bFirstInterface	0x02	First interface number of the set of interfaces that follow this descriptor.
bInterfaceCount	0x02	The Number of interfaces follow this descriptor that are considered "associated".
bFunctionClass	0x02	Class code (assigned by USB-IF) 0x02 = communications and CDC control See table at end of document
bFunctionSubClass	0x02	bInterfaceSubClass used for the associated interfaces
bFunctionProtocol	0x01	bInterfaceProtocol used for the associated interfaces
iFunction	0x00	Index of string descriptor describing the associated interfaces.

[Back to top](#)

Interface 2 Control Descriptor		
bLength	0x09	Size of the descriptor in bytes
bDescriptorType	0x04	Descriptor type (0x04 = interface)
bInterfaceNumber	0x02	Identify number for this interface
bAlternateSetting	0x00	Selected alternate setting for interface
bNumEndpoints	0x01	Number of endpoints supported by interface
bInterfaceClass	0x02	Class code (assigned by USB-IF) 0x02 = communications and CDC control See table at end of document
bInterfaceSubClass	0x02	Subclass code (assigned by USB-IF) See table at end of document
bInterfaceProtocol	0x01	Protocol (assigned by USB-IF) See table at end of document
iInterface	0x00	Index of string descriptor describing interface

[Back to top](#)

Header Management Functional Descriptor		
bFunctionalLength	0x05	Size of the descriptor in bytes
bDescriptorType	0x24	Descriptor type (0x24 = CS_INTERFACE)
bDescriptorSubtype	0x00	Subtype code (0x00 = Header Functional Descriptor, which marks the beginning of the concatenated set of functional descriptors for the interface.)
bcdCDC	0x0110	USB Class Definitions for Communications Devices Specification release number (BCD)

[Back to top](#)

Call Management Functional Descriptor		
bFunctionalLength	0x05	Size of the descriptor in bytes
bDescriptorType	0x24	Descriptor type (0x24 = CS_INTERFACE)
bDescriptorSubtype	0x01	Subtype code (0x01 = Call Management Functional Descriptor)
bmCapabilities	0x00	Bit mask indicating supported features
bDataInterface	0x01	Interface number of Data Class interface. Optionally used for call management

[Back to top](#)

Abstract Control Management Functional Descriptor		
bFunctionalLength	0x04	Size of the descriptor in bytes
bDescriptorType	0x24	Descriptor type (0x24 = CS_INTERFACE)
bDescriptorSubtype	0x02	Subtype code (0x02 = Abstract Control Management Functional Descriptor)
bmCapabilities	0x06	Bit mask indicating supported features

[Back to top](#)

CDC Union Functional Descriptor		
bFunctionalLength	0x05	Size of the descriptor in bytes
bDescriptorType	0x24	Descriptor type (0x24 = CS_INTERFACE)
bDescriptorSubtype	0x06	Subclass code (0x06 = Union Functional Descriptor)
bMasterInterface	0x02	Interface number of the Communication class interface
bSlaveInterface0	0x03	First associated data class Interface
bSlaveInterfaceN	0x??	Additional slave interfaces (if any)

[Back to top](#)

Endpoint 4 Interrupt Descriptor		
bLength	0x07	Size of the descriptor in bytes
bDescriptorType	0x05	Descriptor type (0x05 = endpoint)
bEndpointAddress	0x84	The address and direction of the endpoint
bmAttributes	0x03	Type of transfer: 2 = Bulk, 3 = Control Synchronisation and usage type for isochronous endpoint
wMaxPacketSize	0x0008	Maximum packet size 08 to 64 byte Additional transactions for high speed interrupt and isochronous endpoints
bInterval	0x10	Polling interval for interrupt and isochronous endpoints Maximum NAK rate for bulk and control OUT endpoints

[Back to top](#)

Data Interface 3 Descriptor		
bLength	0x09	Size of the descriptor in bytes
bDescriptorType	0x04	Descriptor type (0x04 = interface)
bInterfaceNumber	0x03	The number of this interface. Zero based value identifying the index in the array of concurrent interfaces supported by this configuration
bAlternateSetting	0x00	Value used to select an alternate setting for the interface identified in the prior field. Allows an interface to change the settings on the fly
bNumEndpoints	0x02	Number of endpoints used by this interface (excluding endpoint 0)
bInterfaceClass	0x0A	Class code (assigned by USB-IF) 0x0A CDC data class See table at end of document
bInterfaceSubClass	0x00	Subclass code (assigned by USB-IF) See table at end of document
bInterfaceProtocol	0x00	Protocol code (assigned by USB-IF) See table at end of document
iInterface	0x00	Index of string descriptors describing the interface

[Back to top](#)

Endpoint 5 IN Descriptor		
bLength	0x07	Size of the descriptor in bytes
bDescriptorType	0x05	Descriptor type (0x05 = endpoint)
bEndpointAddress	0x85	bit 0..3 endpoint number bit 4..6 reserved. Set to 0 bit 7 direction 0=out 1=in
bmAttributes	0x02	bit 0..1 Transfer type 00=Control 01=isochronous 10=bulk 11=interrupt bit 2..7 reserved. If isochronous endpoint bit 3..2 synchronisation type (iso mode) 00=no synchronisation 01=asynchronous 10=adaptive 11=synchronous bit 5..4 usage type (iso mode) 00=data endpoint 01=feedback endpoint 10=explicit feedback data endpoint 11=reserved
wMaxPacketSize	0x0040	Maximum packet size this endpoint is capable of sending or receiving
bInterval	0x00	Interval for polling endpoint data transfers. Value in frame counts. Ignored for bulk and control endpoints. Isochronous must equal 1 and field may range from 1 to 255 for interrupt endpoints

[Back to top](#)

(See [endpoint 5 IN](#) for field info)

Endpoint 6 OUT Descriptor		
bLength	0x07	Size of the descriptor in bytes
bDescriptorType	0x05	Descriptor type (0x05 = endpoint)
bEndpointAddress	0x06	
bmAttributes	0x02	
wMaxPacketSize	0x0040	
bInterval	0x00	

[Back to top](#)

The value of setup fields in more detail.

String descriptors provide human readable information and are optional. If they are not used, any string index fields of descriptors must be set to zero indicating there is no string descriptor available.

The String Descriptor Zero is used to set the languages. The host should read this descriptor to determine what languages are available. If a language is supported, it can then be referenced by sending the language ID in the wIndex field of a [Get Descriptor\(String\)](#) request.

String descriptors		
bLength	0x??	Size of descriptor in bytes
bDescriptorType	0x03	Type of descriptor (0x03 string)
wLangID[0]	0x0409	Language code zero (e.g. 0x0409 English – United States)
wLangID[i]	0x0C09	Language code one (e.g. 0x0C09 English - Australian)
...		
wLangID[x]	0x0407	Language Code x (e.g. 0x0407 German - Standard)

[Back to top](#)

A string descriptor		
bLength	0x03	Size of descriptor in bytes
bDescriptorType	0x03	Type of descriptor (0x03 = string)
bString	string	Unicode encoded string

[Back to top](#)

Each descriptor starts with a bLength and a bDescriptorType field. This makes it possible to concatenate descriptors. Underneath the code of the bDescriptorType field is explained.

bDescriptorType	Descriptor type
0x01	device
0x02	configuration
0x03	String (optional)
0x04	interface
0x05	endpoint
0x06	Device qualifier (only for devices that support both full and high speeds)
0x07	Other speed configuration (only for devices that support both full and high speeds)
0x08	Interface power (not required)
0x09	OTG (only for on the go devices)
0x0A	Debug (not required)
0x0B	Interface association (only for composite devices)
0x24	CS_INTERFACE

[Back to top](#)

If the bDescriptorType is 0x24 CS_INTERFACE it is followed by a bDescriptorSubtype. See underneath for the explanation of the code.

bDescriptorSubtype	Functional description
0x00	Header Functional Descriptor, which marks the beginning of the concatenated set of functional descriptors for the interface.
0x01	Call Management Functional Descriptor.
0x02	Abstract Control Management Functional Descriptor.
0x03	Direct Line Management Functional Descriptor.
0x04	Telephone Ringer Functional Descriptor.
0x05	Telephone Call and Line State Reporting Capabilities Functional Descriptor.
0x06	Union Functional Descriptor
0x07	Country Selection Functional Descriptor
0x08	Telephone Operational Modes Functional Descriptor
0x09	USB Terminal Functional Descriptor
0x0A	Network Channel Terminal Descriptor
0x0B	Protocol Unit Functional Descriptor
0x0C	Extension Unit Functional Descriptor
0x0D	Multi-Channel Management Functional Descriptor
0x0E	CAPi Control Management Functional Descriptor
0x0F	Ethernet Networking Functional Descriptor
0x10	ATM Networking Functional Descriptor
0x11	Wireless Handset Control Model Functional Descriptor
0x12	Mobile Direct Line Model Functional Descriptor
0x13	MDLM Detail Functional Descriptor
0x14	Device Management Model Functional Descriptor
0x15	OBEX Functional Descriptor
0x16	Command Set Functional Descriptor
0x17	Command Set Detail Functional Descriptor
0x18	Telephone Control Model Functional Descriptor
0x19	OBEX Service Identifier Functional Descriptor
0x1A	NCM Functional Descriptor
0x1B	MBIM Functional Descriptor
0x1C	Extended MBIM Functional Descriptor
0x1D	NCM Extended Capability Descriptor
0x1E	NCM Extended Feature Descriptor
0x1F - 0x7F	RESERVED (future use)
0x80 - 0xFE	RESERVED (vendor specific)

[Back to top](#)

Base class

In some of the descriptors the fields bInterfaceClass, bInterfaceSubClass and bInterfaceProtocol are used. The possible combination of these codes is explained underneath.

Base Class	Sub class	Protocol	Descriptor Usage	Description
0x00	0x00	0x00	Device	Use class information in the Interface Descriptors
0x01	0x??	0x??	Interface	Audio
0x02	0x??	0x??	Both	Communications and CDC Control
0x03	0x??	0x??	Interface	HID (Human Interface Device)
0x05	0x??	0x??	Interface	Physical
0x06	0x01	0x01	Interface	Image
0x07	0x??	0x??	Interface	Printer
0x08	0x??	0x??	Interface	Mass Storage
0x09	0x00	0x00 0x01 0x02	Device	Hub
0x0A	0x??	0x??	Interface	CDC-Data
0x0B	0x??	0x??	Interface	Smart Card
0x0D	0x00	0x00	Interface	Content Security
0x0E	0x??	0x??	Interface	Video
0x0F	0x??	0x??	Interface	Personal Healthcare
0x10	0x01 0x02 0x03	0x00 0x00 0x00	Interface	Audio/Video Devices
0x11	0x00	0x00	Device	Billboard Device Class
0x12	0x00	0x00	Interface	USB Type-C Bridge Class
0x13	0x00	0x00	Interface	USB Bulk Display Protocol Device Class
0x14	0x00 0x01	0x01 0x02 0x01 0x02	Interface	MCTP over USB Protocol Endpoint Device Class
0x3C	0x00	0x00	Interface	I3C Device Class
0xDC	0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08	0x01 0x00 0x01 0x00 0x01 0x00 0x01 0x00 0x01 0x00 0x01 0x00 0x01 0x00	Both	Diagnostic Device
0xE0	0x01 0x02	0x01 0x02 0x03 0x04 0x01 0x02 0x03	Interface	Wireless Controller
0xEF	0x01 0x02 0x03 0x04	0x01 0x02 0x01 0x02 0x01 0x01	Both	Miscellaneous

	0x05 0x06 0x07	0x02 0x03 0x04 0x05 0x06 0x07 0x00 0x01 0x02 0x01 0x02 0x01 0x01 0x02		
0xFE	0x01 0x02 0x03	0x01 0x00 0x00 0x01	Interface	Application Specific
0xFF	0x??	0x??	Both	Vendor Specific

[Back to top](#)