USB Signal Action and Measurement Practice (I)

USB Structure, Action Process and Data Transaction Communication Protocol

Preface

Universal Serial Bus, USB for short, is widely applied in peripheral devices of PC/NB and input/output of external devices of mobile devices. For example: the human interface device of PC, keyboard, mouse, flash disk (USB memory stick, pen storage drive, flash drive, etc.), various video/audio input/output devices (webcam, microphone, DVB-T receiver, etc.) and other communication devices and external devices (Bluetooth, infrared, DVD burner, etc.). It even replaces the serial port of old communication (RS232/422/485, PS/2, etc.), and becomes the basic equipment of current PCs.

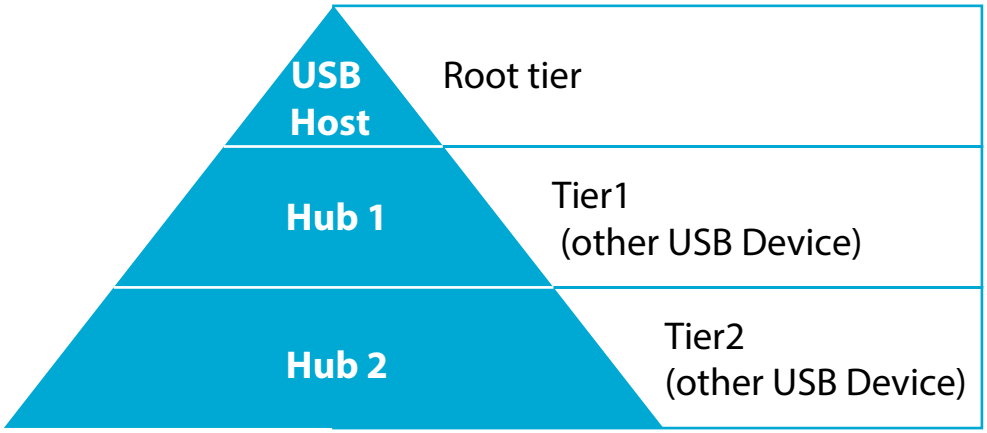
▲ **Fig. 1 : ZEROPLUS LAP-C Series (Logic Cube) & USB2.0 Protocol Analyzer Module**

www.zeroplus.com.tw TEL:+886 2-66202225 FAX:+886 2-22234362

USB Structure and Action Process

According to the rules made by USB Association, the USB structure is divided into Host and Device. Take home computer for an example, its USB Port is USB Host, through this port we could connect many USB devices, and such external device is USB Device.

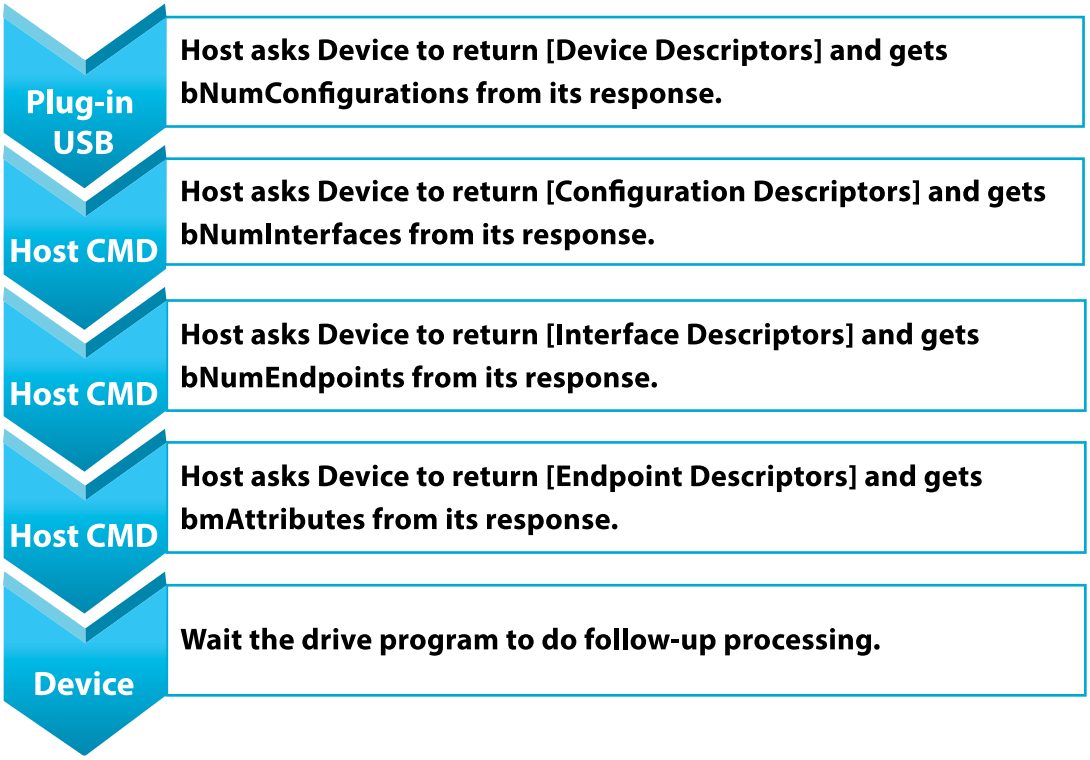
In the bottom communication of PC, USB Host controls the USB communication; it would scan all devices under USB structure through polling mechanism to see if any device joins or any data needs to be returned. No matter the data is transferred from Device to Host, or Host to Device, all USB Devices shall act under the control of USB Host. Below is a simple structure diagram.



▲ **Fig. 2 : USB Communication Structure**

▲ **Fig. 3 : USB Polling Mechanism**

How a USB Device communicates with the Host? The initial flow of USB Device is very important. One by one every device will describe its specification, mode, interface and other crucial information to the USB Host in order to let it recognize and control them without difficulty. Fig. 4 is a basic initial flow. Of course under this system structure, lots of OS need to do USB Device recognition and lots of data need to be acquired, here we will not describe in details.

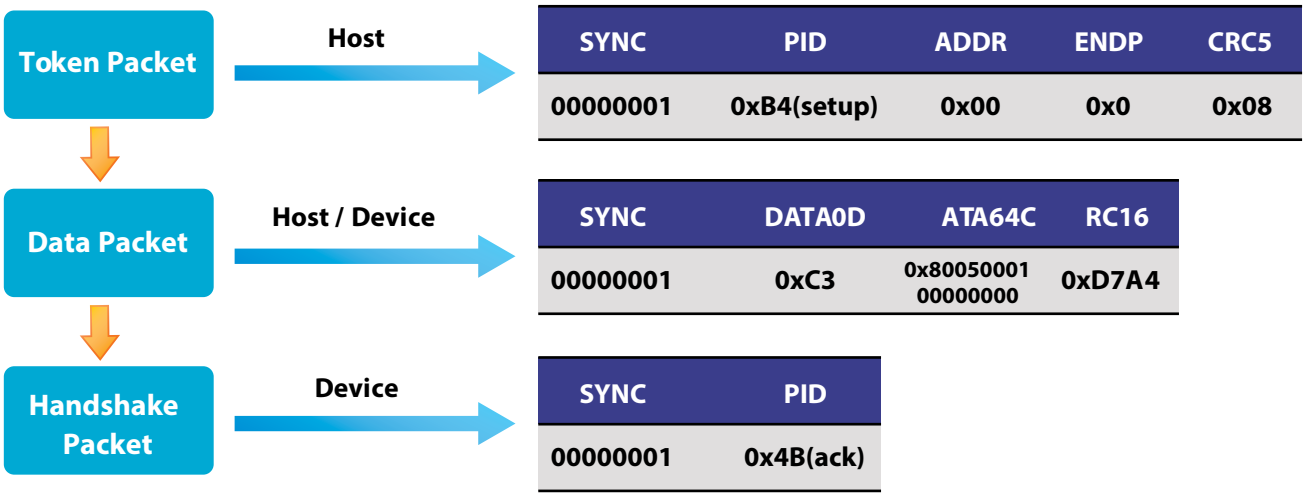


▲ Fig. 4 : USB Device Initial Flow

www.zeroplus.com.tw TEL:+886 2-66202225 FAX:+886 2-22234362

Communication Protocol of USB Interface – Data Transaction

Fig. 5 shows the basic packet structure of USB for data exchange. Firstly the Host will deliver Token Packet to determine the transfer type of transaction, then deliver Data Packet to determine the data type for receiving and transfer direction, then wait the Device to return Handshake Packet to confirm the reception. The signal captured by USB 2.0 protocol analyzer module of ZEROPLUS could also reflect the basic structure of USB communication protocol (Fig. 6).



▲ Fig. 5 : USB Basic Packet Structure

Packet #	Setup	Address	Endpoint	CRC5	EOP	DESCRIBE
16	B4	00	0	08	EOP	SETUP FORMAT

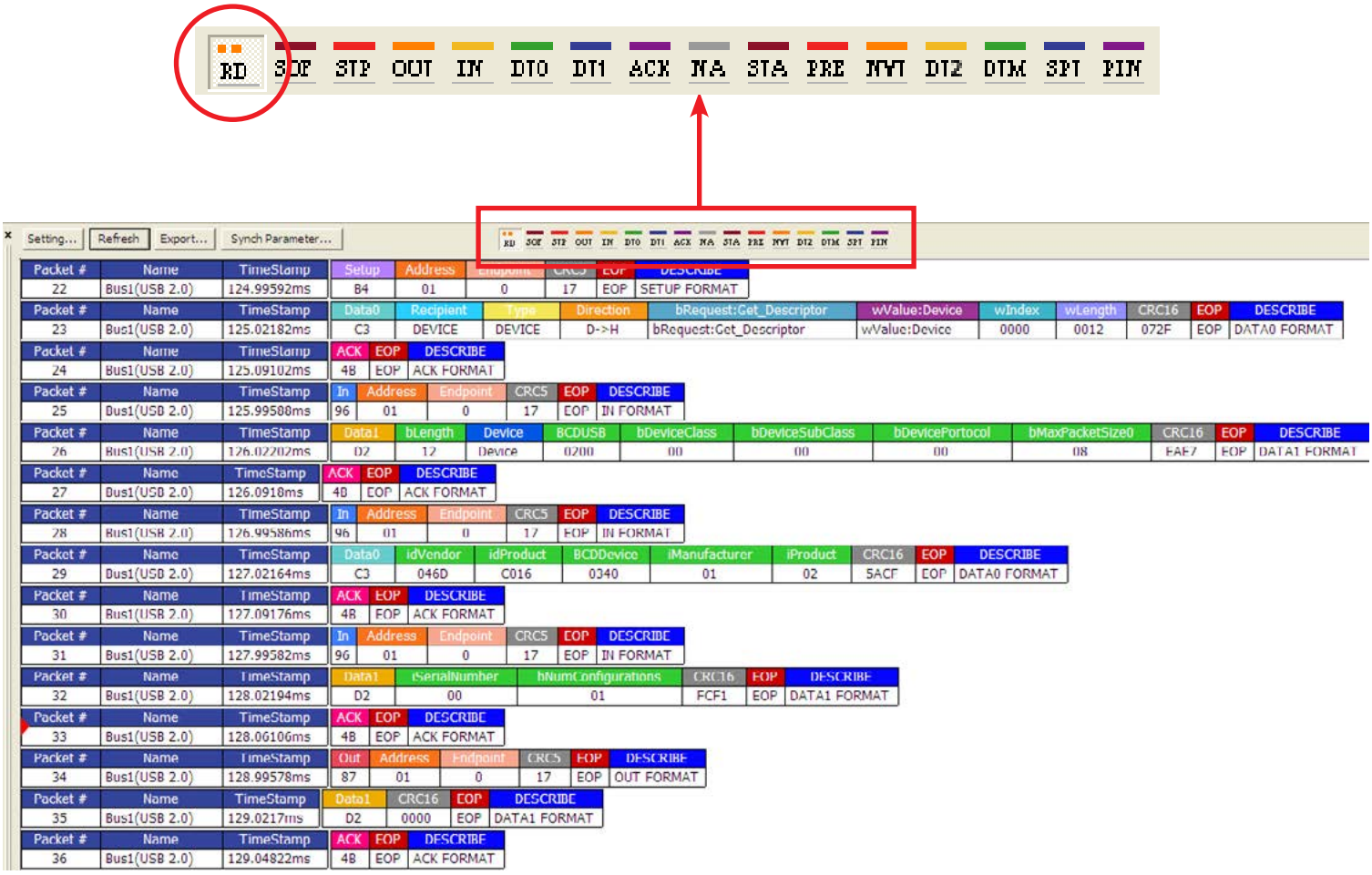
Packet #	Data0	Recipient	Type	Direction	bRequest	wValue	wIndex	wLength	CRC16	EOP	DESCRIBE
17	C3	DEVICE	STANDARD	H->D	05	0001	0000	0000	D7A4	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
18	4B	EOP	ACK FORMAT

▲ Fig. 6 : Packet Data Acquired by USB2.0 Module

USB Mouse Analysis by ZEROPLUS Logic Analyzer

Below is USB1.0 signal data captured by LA at the moment the USB Mouse being plugged into the PC. During USB signal analyzing, since the packets are varied and their structures are complex, what engineers want to see is detailed and intuitive analyzing. To meet this requirement, ZEROPLUS LA supports Request Descriptor decoding, which could further decode the request descriptors of USB Device (such as GetStatus, SetAddress, GetDescriptor, etc.), and this could let engineers analyze the signal much fast. See Fig. 7, in the red box is RD decoding, clicking icons could filter some packet.



▲ **Fig. 7 : Original USB Signal Packet Decoding Status**

USB Device Requests

Standard USB Device Requests

Request	R/V	Function
GetStatus	00H	Read the status of device, port or endpoint.
ClearFeature	01H	Clear or prohibit some character of device, port of endpoint.
SetFeature	03H	Set or enable some character of device, port or endpoint.
SetAddress	05H	Allocate device address.
GetDescriptor	06H	Read specified descriptor.
SetDescriptor	07H	Update existing descriptor or add new descriptor.
GetConfiguration	08H	Read the current configuration value of USB device.
SetConfiguration	09H	Select a suitable configuration for USB device.
GetInterface	0AH	Read the replaceable value of specified port.
SetInterface	0BH	Select a suitable replaceable value for specified port.
SynchFrame	0CH	Read the specified Frame SN of sync port.

Now how to use ZEROPLUS LA and USB2.0 module to verify USB signal? First we must know that one standard USB request packet includes 5 important fields, total 8 bytes. See below image.

USB Device Request Descriptor - Structure

1 Byte	2 Byte	2 Byte	2 Byte	2 Byte
bmRequestType	bRequest	wValue	wIndex	wLength

bmRequestType - Characteristics of request (1 Byte)

D7	D6	D5	D4	D3	D2	D1	D0

Bit 7 : Data Transfer Direction

- 0b = Host-to-device
- 1b = Device-to-host

Bit 6 ... Bit 5 : Request Type

- 00b = Standard
- 01b = Class
- 10b = Vendor
- 11b = Reserved

Bit 4 ... Bit 0 : Recipient

- 00000b = Device
- 00001b = Interface
- 00010b = Endpoint
- 00011b = Other
- 00100b to 11111b = Reserved

bRequest - Specific Request

Please refer to Table Standard USB Device Requests

wValue - Descriptor Type and Descriptor Index of request (2 Byte)

High Byte – 8bit	Low Byte – 8bit
Descriptor Type	Descriptor Index

High Byte (Bit16...Bit8)

- 1 = Device
- 2 = Configuration
- 3 = String
- 4 = Interface
- 5 = Endpoint
- 6 = Device Qualifier
- 7 = Other Speed Configuration
- 8 - Interface Power
- 9 - On-The-Go (OTG)
- 21 = HID (Human interface device)
- 22 = Report
- 23 = Physical

www.zeroplus.com.tw TEL:+886 2-66202225 FAX:+886 2-22234362

wIndex - Index or offset of request (2 Byte)

It is mainly for the Recipient's direction of bmRequestType and description of Index/offset. Below we will briefly introduce Endpoint and Interface.

Endpoint as the recipient for wIndex

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
Direction	Reserved (reset to zero)			Endpoint Number			
D15	D 14	D13	D 12	D11	D 10	D 9	D 8
Reserved (reset to zero)							

Interface as the recipient for wIndex

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
Interface Number							
D15	D 14	D13	D 12	D11	D 10	D 9	D 8
Reserved (reset to zero)							

wLength - Descriptor or byte are Length of request (2 Byte)

According to different situation of bmRequestType, different length would be returned, but two rules must be obeyed:

- If this value is not 0, and the data transfer direction is Device to Host, then this value is not limited.
- If this value is not 0 and the data transfer direction is Host to Device, then this value must meet the data length transferred from Host to Device, or the Device would consider it as abnormal action.

www.zeroplus.com.tw TEL:+886 2-66202225 FAX:+886 2-22234362

USB Mouse Signal Tracking by Sample Application

After completely understand the request forms and functions of USB, next we will show the processing flow of mouse connecting PC and use ZEROPLUS LA to analyze and verify. The processing flow is divided into 4 parts: Get Address, Get Descriptor, Configuration and Set Configuration.

Standard Device Requests – Device of SetAddress

Firstly the USB Host must give the new Device an address, so it delivers Set Address to allocate an address.

Packet #	Setup	Address	Endpoint	CRC5	EOP	DESCRIBE
16	B4	00	0	08	EOP	SETUP FORMAT

Packet #	Data0	Recipient	Type	Direction	bRequest	wValue	wIndex	wLength	CRC16	EOP	DESCRIBE
17	C3	DEVICE	STANDARD	H->D	05	0001	0000	0000	D7A4	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
18	4B	EOP	ACK FORMAT

Pactet#16-#18
The Host delivers bRequest type through packet #17, and the Device would respond its reception of that. If we don't click RD icon to further decode the request, then only value data would be displayed.

Packet #	Setup	Address	Endpoint	CRC5	EOP	DESCRIBE
16	B4	00	0	08	EOP	SETUP FORMAT

Packet #	Data0	Recipient	Type	Direction	bRequest:Set_Addresss	wValue	wIndex	wLength	CRC16	EOP	DESCRIBE
17	C3	DEVICE	DEVICE	H->D	bRequest:Set_Address	0001	0000	0000	D7A4	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
18	4B	EOP	ACK FORMAT

Packet #16 - #18 decode descriptor
Through decoding the descriptor, we know this packet is to Set Address. The first data 0x80 05 00 01 00 00 00 00 measured by general monitoring software (such as Bus Hound) is Data Packet #17. Through RD, we could further analyze Recipient + Type + H-D(direction)=0x80, bRequest=0x05=Set_Address, and wValue=00 01=Device address (give an address number to Device), the following 4 byte is to specify wIndex and wLength.

Standard Device Requests – Device of GetDescriptor

After the USB Host has allocated an address to the Device, it will deliver Get Descriptor to get information about manufacture ID, USB version, type code, etc.. Next we will explain packet format and data in detail, and show the difference between having and not having RD function.

#22 Token Packet

Packet #	Setup	Address	Endpoint	CRC5	EOP	DESCRIBE
22	B4	01	0	17	EOP	SETUP FORMAT

Packet #22 Token packet

The Host delivers the transfer type of Transaction, Device address 0x01 and type setup. This packet is the most basic Token.

#23 Data Packet

Packet #	Data0	Recipient	Type	Direction	bRequest	wValue	wIndex	wLength	CRC16	EOP	DESCRIBE
23	C3	DEVICE	STANDARD	D->H	06	0100	0000	0012	072F	EOP	DATA0 FORMAT

#23 Data Packet

Packet #	Data0	Recipient	Type	Direction	bRequest:Get_Descriptor	wValue:Device	wIndex	wLength	CRC16	EOP	DESCRIBE
23	C3	DEVICE	DEVICE	D->H	bRequest:Get_Descriptor	wValue:Device	0000	0012	072F	EOP	DATA0 FORMAT

Packet #23 Data Packet

The Host asks the Device to give its wanted data. We can see D-H in orange, that means the data is transferred from Device to Host. The request type is bRequest(06) followed by wLength, which is 0x0012(18 byte) in Hexadecimal. When the descriptor RD is decoded, we can clearly see the command bRequest(06) = Get_Descriptor ; wValue specified the descriptor type (Device).

For USB, its direction is LSB to LSB, so on some monitoring software (such as Bus Hound) we may see:
80 06 00 01 00 00 12 00
Here, we change the direction to MSB to LSB for easy analyzing, and directly analyze the meaning of value and display them separately:
01 06 0100 0000 0012
For 80 this byte, we divide it into 5bit-2bit-1bit, and explain them to recipient-type-direction (D-H or H-D).

Handshake Packet

Packet #	ACK	EOP	DESCRIBE
24	4B	EOP	ACK FORMAT

Packet #24 Handshake Packet

The Device returns this packet to the Host to confirm the receiving of above data (#22 and #23) and get the data ready according to the bRequest command of Host.

Next the Device would return data to the Host, for which the data is In Format data, which is to response the Get Descriptor command. Below is the analysis result.

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
25	96	01	0	17	EOP	IN FORMAT

Packet #	Data1	Data	CRC16	EOP	DESCRIBE
26	D2	12 01 00 02 00 00 00 08	EAE7	EOP	DATA1 FORMAT

Packet #	ACK	EOP	DESCRIBE
27	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
28	96	01	0	17	EOP	IN FORMAT

Packet #	Data0	Data	CRC16	EOP	DESCRIBE
29	C3	6D 04 16 C0 40 03 01 02	5ACF	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
30	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
31	96	01	0	17	EOP	IN FORMAT

Packet #	Data1	Data	CRC16	EOP	DESCRIBE
32	D2	00 01	FCF1	EOP	DATA1 FORMAT

Packet #	ACK	EOP	DESCRIBE
33	4B	EOP	ACK FORMAT

Packet #25-#33

From the right image we can know that there are three data transactions: #25, #28 and #31. They are all the data the Device responses to the Host bRequest.

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
25	96	01	0	17	EOP	IN FORMAT

Packet #	Data1	bLength	Device	BCDUSB	bDeviceClass	bDeviceSubClass	bDevicePortocol	bMaxPacketSize0	CRC16	EOP	DESCRIBE
26	D2	12	Device	0200	00	00	00	08	EAE7	EOP	DATA1 FORMAT

Packet #	ACK	EOP	DESCRIBE
27	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
28	96	01	0	17	EOP	IN FORMAT

Packet #	Data0	idVendor	idProduct	BCDDevice	iManufacturer	iProduct	CRC16	EOP	DESCRIBE
29	C3	046D	C016	0340	01	02	5ACF	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
30	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
31	96	01	0	17	EOP	IN FORMAT

Packet #	Data1	iSerialNumber	bNumConfigurations	CRC16	EOP	DESCRIBE
32	D2	00	01	FCF1	EOP	DATA1 FORMAT

Packet #	ACK	EOP	DESCRIBE
33	4B	EOP	ACK FORMAT

Packet #25 - #33 decode descriptor

From above image we can know that, through RD decoding descriptor, the data in Data is completely decoded. The first data of #26 is 12 (Hexadecimal), which has responded the #22 packet wLength = 0x0012 delivered by the Host; the second data Device(01) has responded wValue (descriptor type) of #22 packet. What is much important is that bMaxPacketSize = 08, which means a single packet data only can transfer 8 byte at most. Other code parameters are: bcdUSB=USB version (0200); idVendor=manufacture code (046D); idProduct=product code (C016); bcdeDevice=device BCD code (0340); bDeviceClass=device function class (00); bDeviceSubClass=subclass code (00); bDeviceProtocol=protocol code (00), etc.. The total length of returned data is 0x0012 (18 byte), that echoes what we said before, the length must be accordance with the data.

Standard Device Requests – Device of Configuration

After giving an address to the Device and getting its basic information, the Host would deliver Get-Descriptor command, in which wValue is configuration that used to order the Device to return important property parameters. Below is the whole configuration flow.

Packet #	Setup	Address	Endpoint	CRC5	EOP	DESCRIBE
37	B4	01	0	17	EOP	SETUP FORMAT

Packet #	Data0	Recipient	Type	Direction	bRequest	wValue	wIndex	wLength	CRC16	EOP	DESCRIBE
38	C3	DEVICE	STANDARD	D->H	06	0200	0000	0009	7520	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
39	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
40	96	01	0	17	EOP	IN FORMAT

Packet #	Data1	Data	CRC16	EOP	DESCRIBE
41	D2	09 02 22 00 01 01 00 A0	5019	EOP	DATA1 FORMAT

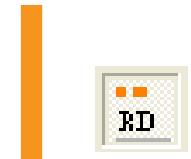
Packet #	ACK	EOP	DESCRIBE
42	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
43	96	01	0	17	EOP	IN FORMAT

Packet #	Data0	Data	CRC16	EOP	DESCRIBE
44	C3	32	8356	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
45	4B	EOP	ACK FORMAT

Packet #37 - #45
From the right image we can know that there would be a 9-byte bRequest(06) command; wValue specifies the descriptor type (0200).



Packet #	Setup	Address	Endpoint	CRC5	EOP	DESCRIBE
37	B4	01	0	17	EOP	SETUP FORMAT

Packet #	Data0	Recipient	Type	Direction	bRequest:Get_Descriptor	wValue:Configuration	wIndex	wLength	CRC16	EOP	DESCRIBE
38	C3	DEVICE	DEVICE	D->H	bRequest:Get_Descriptor	wValue:Configuration	0000	0009	7520	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
39	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
40	96	01	0	17	EOP	IN FORMAT

Packet #	Data1	bLength	Configuration	wTotalLength	bNumInterfaces	bConfigurationValue	iConfiguration	bmAttributes	CRC16	EOP	DESCRIBE
41	D2	09	Configuration	0022	01	01	00	A0	5019	EOP	DATA1 FORMAT

Packet #	ACK	EOP	DESCRIBE
42	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
43	96	01	0	17	EOP	IN FORMAT

Packet #	Data0	MaxPower	CRC16	EOP	DESCRIBE
44	C3	32	8356	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
45	4B	EOP	ACK FORMAT

Packet #37 - #45 After RD decoding.
After RD decoding descriptor, we can know #38 packet is to deliver Get_Descriptor command for the Host and specify descriptor type to configuration, data total length is 9 byte. From #41 we can see the 9 byte data returned from Device to Host are: first, Length=09, fitting the data length delivered by the Host; second, Configuration=02, also fitting the descriptor type needed by the Host; third, 0x22, which means there would be 34 byte data needed to be transferred in the whole configuration; the last one is MaxPower = 0x32*2mA = 100mA.

From above configuration we can know that, the first configuration command delivered by the Host is to get the length of configuration data and special property of Device, the second configuration command is to get the complete configuration property of Device.

Packet #	Setup	Address	Endpoint	CRC5	EOP	DESCRIBE
157	B4	01	0	17	EOP	SETUP FORMAT

Packet #	Data0	Recipient	Type	Direction	bRequest	wValue	wIndex	wLength	CRC16	EOP	DESCRIBE
158	C3	DEVICE	STANDARD	D->H	06	0200	0000	0022	0D2F	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
159	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
160	96	01	0	17	EOP	IN FORMAT

Packet #	Data1	Data	CRC16	EOP	DESCRIBE
161	D2	09 02 22 00 01 01 00 A0	5019	EOP	DATA1 FORMAT

Packet #	ACK	EOP	DESCRIBE
162	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
163	96	01	0	17	EOP	IN FORMAT

Packet #	Data0	Data	CRC16	EOP	DESCRIBE
164	C3	32 09 04 00 00 01 03 01	ACB2	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
165	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
166	96	01	0	17	EOP	IN FORMAT

Packet #	Data1	Data	CRC16	EOP	DESCRIBE
167	D2	02 00 09 21 10 01 00 01	68CE	EOP	DATA1 FORMAT

Packet #	ACK	EOP	DESCRIBE
168	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
169	96	01	0	17	EOP	IN FORMAT

Packet #	Data0	Data	CRC16	EOP	DESCRIBE
170	C3	22 34 00 07 05 81 03 04	B587	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
171	4B	EOP	ACK FORMAT

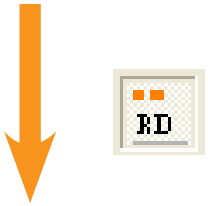
Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
172	96	01	0	17	EOP	IN FORMAT

Packet #	Data1	Data	CRC16	EOP	DESCRIBE
173	D2	00 0A	7E12	EOP	DATA1 FORMAT

Packet #	ACK	EOP	DESCRIBE
174	4B	EOP	ACK FORMAT

Packet #157 - #174
From right image, we can know that the Host has gotten the whole configuration length 0x22 (34 byte), so packet #158 would be delivered like this:
80 06 00 02 00 00 22 00
But here that would be converted automatically to:
01 - 06 – 0200 – 0000 – 0022

Packet #157 - #174 after decoding
Through decoding we can see that the configuration of the Device has 4 types of data:
Configuration - (9Byte)
Interface - (9Byte)
HID - (9Byte)
Endpoint - (7Byte)
Total length is 34 byte, which fitting the data length 0x22 (34 byte) delivered by the Host.
Since our device is mouse (HID), so we can find a property data of bInterfaceClass and code 0x03 in the Interface packet; this code is HID, fitting HID Class.



Packet #	Setup	Address	Endpoint	CRC5	EOP	DESCRIBE
157	B4	01	0	17	EOP	SETUP FORMAT

Packet #	Data0	Recipient	Type	Direction	bRequest:Get_Descriptor	wValue:Configuration	wIndex	wLength	CRC16	EOP	DESCRIBE
158	C3	DEVICE	DEVICE	D->H	bRequest:Get_Descriptor	wValue:Configuration	0000	0022	0D2F	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
159	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
160	96	01	0	17	EOP	IN FORMAT

Packet #	Data1	bLength	Configuration	wTotalLength	bNumInterfaces	bConfigurationValue	iConfiguration	bmAttributes	CRC16	EOP	DESCRIBE
161	D2	09	Configuration	0022	01	01	00	A0	5019	EOP	DATA1 FORMAT

Packet #	ACK	EOP	DESCRIBE
162	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
163	96	01	0	17	EOP	IN FORMAT

Packet #	Data0	MaxPower	bLength	Interface	bInterfaceNumber	bAlternateSetting	bNumEndpoints	bInterfaceClass	bInterfaceSubClass	CRC16	EOP	DESCRIBE
164	C3	32	09	Interface	00	00	01	03	01	ACB2	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
165	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
166	96	01	0	17	EOP	IN FORMAT

Packet #	Data1	bInterfaceProtocol	iInterface	bLength	HID	BCDHID	bCountryCode	bNumDescriptors	CRC16	EOP	DESCRIBE
167	D2	02	00	09	HID	0110	00	01	68CE	EOP	DATA1 FORMAT

Packet #	ACK	EOP	DESCRIBE
168	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
169	96	01	0	17	EOP	IN FORMAT

Packet #	Data0	bDescriptorType	wDescriptorLength	bLength	Endpoint	bEndpointAddress	bmAttributes	wMaxPacketSize	CRC16	EOP	DESCRIBE
170	C3	22	0034	07	End_point	81	03	04	B587	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
171	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
172	96	01	0	17	EOP	IN FORMAT

Packet #	Data1	wMaxPacketSize	bInterval	CRC16	EOP	DESCRIBE
173	D2	00	0A	7E12	EOP	DATA1 FORMAT

Packet #	ACK	EOP	DESCRIBE
174	4B	EOP	ACK FORMAT

Standard Device Requests – SetConfiguration for Device

After getting complete data of Device, the Host would deliver SetConfiguration to set its property parameters and GetInterface.

Packet #	Setup	Address	Endpoint	CRC5	EOP	DESCRIBE
178	B4	01	0	17	EOP	SETUP FORMAT

Packet #	Data0	Recipient	Type	Direction	bRequest	wValue	wIndex	wLength	CRC16	EOP	DESCRIBE
179	C3	DEVICE	STANDARD	H->D	09	0001	0000	0000	E4A4	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
180	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
181	96	01	0	17	EOP	IN FORMAT

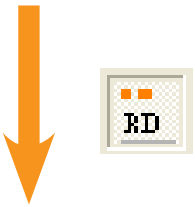
Packet #	Data1	CRC16	EOP	DESCRIBE
182	D2	0000	EOP	DATA1 FORMAT

Packet #	ACK	EOP	DESCRIBE
183	4B	EOP	ACK FORMAT

Packet #	Setup	Address	Endpoint	CRC5	EOP	DESCRIBE
184	B4	01	0	17	EOP	SETUP FORMAT

Packet #	Data0	Recipient	Type	Direction	bRequest	wValue	wIndex	wLength	CRC16	EOP	DESCRIBE
185	C3	INTERFACE	CLASS	H->D	0A	0000	0000	0000	6B04	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
186	4B	EOP	ACK FORMAT



Packet #178 - #186

#178 - #180 are Set Configuration

00 09 01 00 00 00 00 00

#184 - #186 are Get Interface

21 0A 00 00 00 00 00 00

Packet #	Setup	Address	Endpoint	CRC5	EOP	DESCRIBE
178	B4	01	0	17	EOP	SETUP FORMAT

Packet #	Data0	Recipient	Type	Direction	bRequest:Set_Configuration	wValue	wIndex	wLength	CRC16	EOP	DESCRIBE
179	C3	DEVICE	DEVICE	H->D	bRequest:Set_CONFIGURATION	0001	0000	0000	E4A4	EOP	DATA0 FORMAT

Packet #	ACK	EOP	DESCRIBE
180	4B	EOP	ACK FORMAT

Packet #	In	Address	Endpoint	CRC5	EOP	DESCRIBE
181	96	01	0	17	EOP	IN FORMAT

Packet #	Data1	CRC16	EOP	DESCRIBE
182	D2	0000	EOP	DATA1 FORMAT

Packet #	ACK	EOP	DESCRIBE
183	4B	EOP	ACK FORMAT

Packet #	Setup	Address	Endpoint	CRC5	EOP	DESCRIBE
184	B4	01	0	17	EOP	SETUP FORMAT

Packet #	Data0	Recipient	Type	Direction	bRequest:Get_Interface	wValue	wIndex	wLength	CRC16	EOP	DESCRIBE
185	C3	INTERFACE	INTERFACE	H->D	bRequest:Get_Interface	0000	0000	0000	6B04	EOP	DATA0 FORMAT

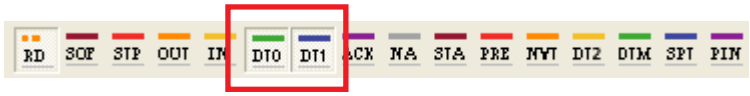
Packet #	ACK	EOP	DESCRIBE
186	4B	EOP	ACK FORMAT

Packet #178 - #186

Through RD complete decoding, we can clearly see that there are two request commands in above image with direction be Host to Device, which means the Host would transfer data to the Device. Packet #179 requests Set Configuration (09), Configuration Value (0001) and wLength (0 byte), but it has no special requirement for the content, so the Device just needs to respond the receiving of packet #182 to the Host. Also, the Get Interface of next command packet #185 has no special requirement.

In the end, the USB Host would deliver Get Descriptor from Device to Host, in order to make the mouse report to the Interface, not Device. The content would be different because of different manufacturer. The reported content includes mouse movements, button clicks, etc..

On the other side, the report length is decided by the designer. In this example, the report length of tested mouse is 52 byte. The last 2 byte are universal value code: 0xC0 and 0xC0, which means the data transfer ends. To see the report data conveniently, in Fig. 7 only packets of Data0 and Data1 are displayed, the others are filtered. Tab. 2 is report content.



Packet #	Data0	Recipient	Type	Direction	bRequest:Get_Descriptor	wValue:Report	wIndex	wLength	CRC16	EOP	DESCRIBE
64	C3	INTERFACE	INTERFACE	D->H	bRequest:Get_Descriptor	wValue:Report	0000	0074	F3FA	EOP	DATA0 FORMAT

Packet #	Data1	Data	CRC16	EOP	DESCRIBE
65	D2	05 01 09 02 A1 01 09 01	C57F	EOP	DATA1 FORMAT

Packet #	Data0	Data	CRC16	EOP	DESCRIBE
66	C3	A1 00 05 09 19 01 29 03	8598	EOP	DATA0 FORMAT

Packet #	Data1	Data	CRC16	EOP	DESCRIBE
67	D2	15 00 25 01 75 01 95 03	04F5	EOP	DATA1 FORMAT

Packet #	Data0	Data	CRC16	EOP	DESCRIBE
68	C3	81 02 75 05 95 01 81 01	F286	EOP	DATA0 FORMAT

Packet #	Data1	Data	CRC16	EOP	DESCRIBE
69	D2	05 01 09 30 09 31 09 38	5EE2	EOP	DATA1 FORMAT

Packet #	Data0	Data	CRC16	EOP	DESCRIBE
70	C3	15 81 25 7F 75 08 95 03	92F5	EOP	DATA0 FORMAT

Packet #	Data1	Data	CRC16	EOP	DESCRIBE
71	D2	81 06 C0 C0	E66D	EOP	DATA1 FORMAT

▲ Fig. 8 : Report of Mouse

Packet #65			Packet #66			Packet #67			Packet #68		
05	01	USAGE_PAGE (Generic Desktop)	A1	00	COLLECTION (Physical)	15	00	LOGICAL_MINIMUM (0)	81	02	INPUT (Data,Var,Abs)
09	02	USAGE (Mouse)	05	09	USAGE_PAGE (Button)	25	01	LOGICAL_MAXIMUM (1)	75	05	REPORT_SIZE (5)
A1	01	COLLECTION (Application)	19	01	USAGE_MINIMUM (Button 1)	75	01	REPORT_SIZE (1)	95	01	REPORT_COUNT (1)
09	01	USAGE (Pointer)	29	03	USAGE_MAXIMUM (Button 3)	95	03	REPORT_COUNT (3)	81	01	INPUT (Cnst,Var,Abs)

Packet #69			Packet #70			Packet #71		
05	01	USAGE_PAGE (Generic Desktop)	15	81	LOGICAL_MINIMUM (-127)	81	06	INPUT (Data,Var,Rel)
09	30	USAGE (X)	25	7F	LOGICAL_MAXIMUM (127)	C0	C0	END_COLLECTION
09	31	USAGE (Y)	75	08	REPORT_SIZE (8)			
09	38	USAGE (Wheel)	95	03	REPORT_COUNT (2)			

▲ Tab. 1 : Mouse Report Table

to be continued

In part II of this article, we will share with you [USB-OTG Actual Measurement Case], in which we would take the measurement of smart phone OTG connecting with peripheral and storage device as an example, to be expected.

For more product information, please go to Zeroplus official website (繁體中文 / 简体中文 / English)

▲ ZEROPLUS Logic Analyzer LAP-C Series

▲ USB 2.0 Protocol Analyzer Module

Note: USB2.0 protocol analyzer module can only be used for models with 32 channels, and only for C Series.

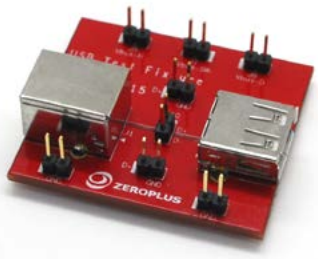
USB 2.0 Protocol Analyzer Module

USB2.0 協定分析模組
USB2.0 协议分析模组
USB 2.0 Capture Board

USB2.0 Protocol Decoding Module

USB2.0 解碼軟體
USB2.0 解码软件
USB2.0 Decoding software

▲ USB 2.0 Capture Board



▲ USB Test Fixture(B TO A)

ZEROPLUS **provides a lot of USB measurement data, please go to " Technical Support "**

- USB 2.0 Signal Analysis Skills
- Debugging USB1.1 Buses in Embedded System Designs

Roger Tsai
Instrument Department/ FAE