usb command

reset command

Chapter 4 Architectual Overview

4.6 System Configuration

4.6.3 Bus Enumeration

Chapter 7 Electrical->

7.1 Signaling->

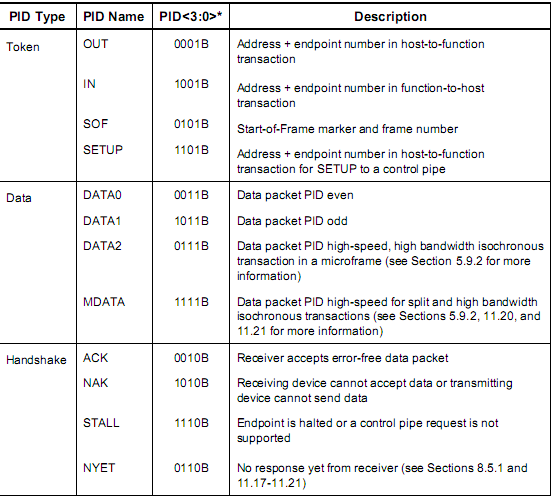
7.1.7 Signaling Levels

7.1.7.5 Reset Signaling

Chapter 8 Protocol Layer

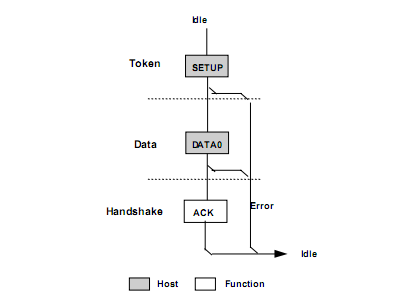
8.3 Packet Field Formats

8.3.1 Packet Identifier Field



8.5 Transaction Packet Sequences

8.5.3 Control transfers



Chapter 9 USB Device Framework->

9.1 USB Device States

9.1.2 Bus Enumeration

When a USB device is attached to or removed from the USB,the host uses a process known as bus

enumeration to identify and manage the device state changes necessary.

1. The hub to which the USB device is now attached informs the host of the event via a reply on its status change pipe.At this point,the USB device is in the Powered state and the port to which it is attached is disabled.
2. The host determines the exact nature of the change by querying the hub.
3. Now that the host knows the port to which the new device has been attached,the host then waits for at least 100ms to allow completion of an insertion process and for power at the device to become stable.**The host then issues a port enable and reset command to that port.** Refer to Section 7.1.7.5 for sequence of events and timings of connection throught device reset.
4. The hub performs the required reset processing for that port(see Section 11.5.1.5).When the reset signal is released,the port has been enabled. The USB device is now in the Default state and can draw no more than 100mA from VBUS.All of its registers and state have been reset and it answers to the default address.
5. The host assigns a unique address to the USB device,moving the device to the Address State
6. Before the USB device receives a unique address,its Default Control Pipe is still accessible via the default address.The host reads the device descriptor to determine what actual maximum data payload size this USB device’s default pipe can use
7. The host reads the configuration information from the device by reading each configuration zero to n-1,where n is the number of configurations.this process may take several milliseconds to complete
8. Based on the configuration information and how the USB device will be used,the host assigns a configuraiton value to the device.the device is now in the Configured state.The USB device may now draw the amount of VBUS power described in its descriptor for the selected configuration.From the device’s point of view,it is now ready for use.

问题1:

usb address 什么时候分配。

hub\_port\_connect

{

int i;

choose\_devnum(udev)

for(i=0;i<SET\_CONFIG\_TRIES;i++) {

udev = usb\_alloc\_dev(hdev,hdev->bus,port1);

usb\_set\_device\_state(udev,USB\_STATE\_POWERED);

status = hub\_port\_init(hub,udev,port1,i);

}

}

hub\_port\_init(hub,udev,port1,i)

{

int devnum =udev->devnum;

/\*

Reset the device;full speed may morph to high speed

FIXME a USB 2.0 device may morph into SuperSpeed on reset

\*/

retval = hub\_port\_reset(hub,port1,udev,delay,false);

if(retval <0)

goto fail;

switch(udev->speed ) {

case USB\_SPEED\_SUPER\_PLUS:

}

for(retries =0;retries < GET\_DESCRIPTOR\_TRIES;(++retries,msleep(100)))

{

bool did\_new\_scheme = false;

if(use\_new\_scheme(udev,retry\_counter,port\_dev))

{

}

}

}

}

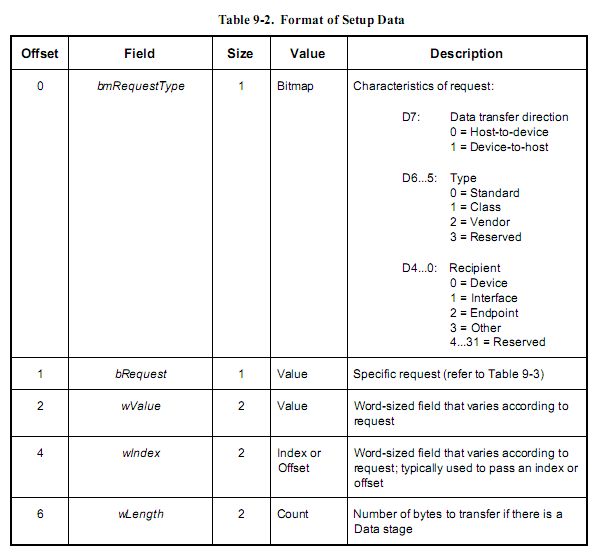
问题2:

usb endpoint 0啥时候不能用了

Chapter 9 USB Device Framework->

9.3 USB Device Requests

ALL USB devices respond to requests from the host on the device’s Default Control Pipe.The request and the request’s parameters are sent to the device in the Setup packet.

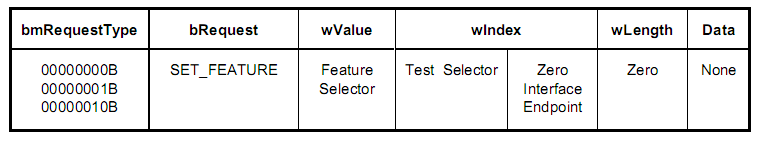




9.4 Standard Device Requests

9.4.9 Set Feature

This request is used to set or enable a specific feature.



Chapter 11 Hub Specification

11.5 Downstream Facing Ports

11.5.1 Downstreaming Facing Port State Descriptions

11.5.1.1 Not Configured

11.5.1.5 Resetting

11.24 Requests

11.24.2 Class-specific Requests

The hub class defines requests to which hubs respond.

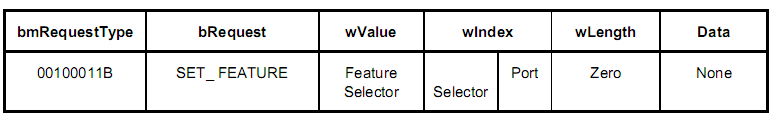
Table 11-17 gives the valid feature selectors for the hub class.

11.24.2.7 Get Port Status

11.24.2.7.1 Port Status Bits

11.24.2.13 Set Port Feature

This request sets a value reported in the port status.



The port number must be a valid port number for that hub,greater than zero. The port number is in the least significant byte of the wIndex field.

Setting a feature enables that feature or starts a process associated with that feature;

see Table 11-17 for the feature selector definitions that apply to a port as a recipient.

features that can be set with this request are:

PORT\_RESET

PORT\_TEST

..

C\_PORT\_RESET\*

Setting the PORT\_SUSPEND feature causes bus traffic to cease on that port and consequently ,the device to suspend.

Setting the reset feature PORT\_RESET causes the hub to signal reset on that port.When the reset signaling is complete,the hub sets the C\_PORT\_RESET status change and immediately enables the port.

When the feature selector is PORT\_TEST,the most significant byte of the wIndex field is the selector identifying the specific test mode. Table 11-24 lists the test selector definitions.

Test mode of a downstream facing port

hub\_port\_init->

hub\_port\_reset(hub,port1,udev,delay,warm)

{

for(i=0;i<PORT\_RESET\_TRIES;i++)

{

status=

set\_port\_feature(hub->hdev,port1,(warm?USB\_PORT\_FEAT\_BH\_PORT\_RESET

:USB\_PORT\_FEAT\_RESET));

}

}

## endpoint descriptor bInterval

<https://docs.microsoft.com/en-us/windows-hardware/drivers/ddi/usbspec/ns-usbspec-_usb_endpoint_descriptor>

the bInterval value contains the polling interval for interrupt and isochronous endpoints.For other types of endpoint,this value should be ignored.

This value reflects the device’s configuration in firmware.

ehci

4. Operational Model

4.10 Managing Control/Bulk/Interrupt Transfers via Queue Heads

4.10.7 Adding Interrupt Queue Heads to the Periodic Schedule

The link path(s) from the periodic frame list

## hub remaining power

struct usb\_hub {

struct usb\_hub\_descriptor \*descriptor;

};